ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

# The Nephroblastoma Oncosimulator:

# Clinical adaptation for patients of distinct histologic profiles using high performance computing

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΗΣ

**ΦΩΤΕΙΝΗΣ ΠΑΝΑΓΙΩΤΙΔΟΥ**

**Επιβλέπων :** Δημήτριος Σούντρης
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2021

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

# The Nephroblastoma Oncosimulator:

# Clinical adaptation for patients of distinct histologic profiles using high performance computing

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΗΣ

**ΦΩΤΕΙΝΗΣ ΠΑΝΑΓΙΩΤΙΔΟΥ**

**Επιβλέπων :** Δημήτριος Σούντρης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 19η Οκτωβρίου 2021.

| | | |
|---|---|---|
| ............................ | ............................ | ............................ |
| Δημήτριος Σούντης | Γεώργιος Σταματάκος | Μιχάλης Καβουσανάκης |
| Καθηγητής Ε.Μ.Π. | Διευθυντής Ερευνών ΕΠΙΣΕΥ | Επίκουρος Καθηγητής Ε.Μ.Π. |

Αθήνα, Οκτώβριος 2021

...................................
Φωτεινή Παναγιωτίδου

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

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

# The Nephroblastoma Oncosimulator:

# Clinical adaptation for patients of distinct histologic profiles using high performance computing

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

της

Φωτεινής Παναγιωτίδου

Αθήνα, Οκτώβριος 2021

**Επίβλεψη:**

Δημήτριος Σούντρης, Καθηγητής ΕΜΠ

Microprocessors and Digital Systems Lab

Γεώργιος Σταματάκος, Διευθυντής Ερευνών ΕΠΙΣΕΥ-ΕΜΠ

In Silico Oncology and In Silico Medicine Group

NATIONAL TECHNICAL UNIVERSITY OF ATHENS

SCHOOL OF ELECTRICAL & COMPUTER ENGINEERING

# The Nephroblastoma Oncosimulator:

# Clinical adaptation for patients of distinct histologic profiles using high performance computing

**DIPLOMA THESIS**

of

Foteini Panagiotidou

Athens, October 2021

**Academic advisors:**

Dimitrios Soudris, NTUA professor

Microprocessors and Digital Systems Lab

Georgios Stamatakos, ICCS-NTUA research professor

In Silico Oncology and In Silico Medicine Group

# Ευχαριστίες

Με την ολοκλήρωση της παρούσας εργασίας, θα ήθελα να ευχαριστήσω όλους όσους με βοήθησαν στην εκπόνησή της. Αρχικά, θέλω να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Δημήτριο Σούντρη, για την διεύρυνση και προώθηση των ενδιαφερόντων μου καθ' όλη τη διάρκεια των σπουδών μου με την εξαιρετική δουλειά που κάνει στο εργαστήριο μικροεπεξεργαστών και ψηφιακών συστημάτων, για την εμπιστοσύνη που μου έδειξε με την ανάθεση της διπλωματικής εργασίας και για την υποστήριξή του σε όλη τη διάρκεια της συνεργασίας μας, καθώς και τον υποψήφιο διδάκτορα Δημοσθένη Μασούρο, ο οποίος συμμετείχε στην επίβλεψη, ήταν πάντα διαθέσιμος για να απαντάει σε απορίες και να λύνει πρακτικά ζητήματα και βοήθησε πολύ με τα εύστοχα σχόλια και τις συμβουλές του. Ακόμα, θέλω να ευχαριστήσω τον κ. Γεώργιο Σταματάκο, διευθυντή ερευνών στο Ερευνητικό Πανεπιστημιακό Ινστιτούτο Συστημάτων Επικοινωνιών και Υπολογιστών, που μου μετέδωσε τον ενθουσιασμό και τις γνώσεις του πάνω στον τομέα της in silico ιατρικής, που με ενέταξε στην άψογη δουλειά που κάνει σε αυτό το χώρο και που συμμετείχε στην επίβλεψη της εργασίας με ιδέες, δημιουργικά σχόλια και παρατηρήσεις, καθώς και τους συνεργάτες του στην In Silico Oncology and In Silico Medicine Group, Ελένη Γεωργιάδη, Ελένη Κολοκοτρώνη και Νικόλαο Χριστοδούλου, που ήταν πάντα διαθέσιμοι να δώσουν συμβουλές για πρακτικά ζητήματα του Ογκοπροσομοιωτή. Επίσης, πολύτιμη ήταν η συμβολή του καθηγητή Norbert Graf και του Marcel Meyerheim, οι οποίοι ετοίμασαν και μοιράστηκαν μαζί μας τα ιατρικά δεδομένα των ασθενών από το πανεπιστημιακό νοσοκομείο του Saarland, χωρίς τα οποία δεν θα ήταν δυνατή η ολοκλήρωση της εργασίας. Τέλος, ευχαριστώ απεριόριστα την οικογένεια και τους φίλους μου, που με υποστηρίζουν πάντα, που πιστεύουν στις δυνάμεις μου και μου δίνουν θάρρος.

# Περίληψη

Ο Ογκοπροσομοιωτής είναι ένα διακριτό top-down μοντέλο για την πολυκλιμακωτή προσομοίωση της ασθένειας του καρκίνου, που απεικονίζει τον όγκο in silico σε ένα τρισδιάστατο πλέγμα από voxels και χρησιμοποιεί τη στρατηγική “summarize and jump” για την προσομοίωση της χρονικής του εξέλιξής στο κυτταρικό και ιστικό επίπεδο βιοπολυπλοκότητας. Ο Ογκοπροσομοιωτής Νεφροβαστώματος είναι μια εκδοχή του Ογκοπροσομοιωτή, ειδική για την in silico απεικόνιση του όγκου του Wilms, ο οποίος είναι ο συχνότερος νεφρικός όγκος στην παιδική ηλικία. Αναπτύχθηκε από μέλη της Ομάδας για την In Silico Ογκολογία και την In Silico Ιατρική των ΕΠΙΣΕΥ, ΣΗΜΜΥ, ΕΜΠ, υπό την επιστημονική καθοδήγηση του Γ. Σταματάκου και την κλινική εποπτεία του N. Graf, Πανεπιστήμιο του Saarland, από το 2006 έως το 2019. Ο απώτερος στόχος του Ογκοπροσομοιωτή και των ειδικών ανά ασθένεια εκδοχών του είναι να λειτουργούν ως ιατρικοί σύμβουλοι για εξατομικευμένη ιατρική δημιουργώντας ένα digital twin για τον όγκο και τον φυσιολογικό ιστό που τον περιβάλλει. Ωστόσο, για το σκοπό αυτό χρειάζεται η μετάφραση στον υπολογιστή των ιατρικών δεδομένων για το συγκεκριμένο ασθενή και, παρόλο που αυτό γίνεται εύκολα για τα απεικονιστικά δεδομένα και το μέγεθος και το σχήμα του όγκου, οι τιμές των παραμέτρων εισόδου της προσομοίωσης που καθορίζουν τις προσομοιούμενες βιολογικές διεργασίες δεν ορίζονται μονοσήμαντα από τα ιατρικά δεδομένα. Ο σκοπός της παρούσας εργασίας είναι να αντιμετωπίσει αυτό το ζήτημα διερευνώντας μεθόδους high performance computing για τη βελτίωση της απόδοσης μιας μεμονωμένης προσομοίωσης και την αποδοτική εκτέλεση πολλαπλών προσομοιώσεων εικονικών ασθενών ανά φυσικό ασθενή, με κάθε εικονικό ασθενή να έχει ξεχωριστή τιμή για την από κοινού κατανομή των παραμέτρων εισόδου. Η ομάδα προσομοιώσεων εικονικών ασθενών εκτελέστηκε για τρεις ασθενείς που έχουν ήδη δεχθεί θεραπεία και που εκπροσωπούν διακριτά ιστολογικά προφίλ και αντίστοιχες ομάδες κινδύνου, τα ιατρικά δεδομένα των οποίων παρέχονται από το πανεπιστημιακό νοσοκομείο του Saarland, με κάθε εικονικό ασθενή να έχει μια ξεχωριστή τιμή για την από κοινού κατανομή των παραμέτρων εισόδου του μοντέλου και την τιμή για την παράμετρο CKR, που εκφράζει τα αποτελέσματα της χορηγούμενης θεραπείας, να προσαρμόζεται για το καθένα εικονικό ασθενή μέχρι τα αποτελέσματα της προσομοίωσης να ταιριάζουν με τα πραγματικά αποτελέσματα του φυσικού ασθενή. Πριν από την εκτέλεση των προσομοιώσεων εικονικών ασθενών που υλοποιούν την in silico κλινική προσαρμογή της παραμέτρου CKR, το μέγεθος του voxel του τρισδιάστατου πλέγματος προσαρμόστηκε για τις εισόδους που αντιστοιχούν σε κάθε ασθενή του συνόλου δεδομένων στο πλαίσιο ενός βήματος προεπεξεργασίας δεδομένων που διαχειρίζεται το tradeoff ανάλυσης και κόστους της προσομοίωσης, εφαρμόστηκαν βελτιστοποιήσεις στον κώδικα για την αξιοποίηση βελτιωμένων πόρων και την ταυτόχρονη εκτέλεση υπολογισμών όπου οι εξαρτήσεις δεδομένων το επιτρέπουν και πραγματοποιήθηκε ανάλυση ευαισθησίας για τον έλεγχο της προεπεξεργασίας των δεδομένων εισόδου και του βελτιστοποιημένου πηγαίου κώδικα μέσω της σύγκρισης με προηγούμενες verified εκτελέσεις προσομοιώσεων. Το βήμα προεπεξεργασίας δεδομένων εισάγει σημαντικό speedup στο χρόνο εκτέλεσης της προσομοίωσης (περίπου $37-688\%$) και μείωση στο memory footprint της εκτέλεσης (περίπου $87-94\%$), ενώ έμμεσα βελτιώνει τις προοπτικές για speedup από το βήμα βελτιστοποίησης του πηγαίου κώδικα, σύμφωνα με το νόμο του Amdhal, καθιστώντας την ισορροπία φόρτου εργασίας της προσομοίωσης πιο δίκαιη και λιγότερο έντονη στα τμήματα που δεν ωφελούνται από τους βελτιωμένους πόρους. Η βελτιστοποίηση του πηγαίου κώδικα εισάγει speedup περίπου $16-24\%$ εστιάζοντας στην εκτέλεση παράλληλων υπολογισμών για το χωρίς εξαρτήσεις δεδομένων τρίτο σκανάρισμα που ολοκληρώνει την προσομοίωση στο κυτταρικό επίπεδο βιοπολυπλοκότητας. Το τελευταίο βήμα της κλινικής προσαρμογής υλοποιήθηκε για 20 και 200 εικονικούς ασθενείς ανά φυσικό ασθενή του συνόλου δεδομένων, παράγοντας μία διακριτή κατανομή, δηλαδή μέση τιμή και τυπική απόκλιση, για την παράμετρο CKR για κάθε ξεχωριστή ομάδα κινδύνου του συνόλου δεδομένων.

**Λέξεις κλειδιά:** Ογκοπροσομοιωτής, Ογκοπροσομοιωτής Νεφροβλαστώματος, Όγκος του Wilms, In silico ιατρική, Πολυκλιμακωτή προσομοίωση, Digital twin, High performance computing, Valgrind, OpenMP, Εικονικοί ασθενείς, Κλινική προσαρμογή

# Acknowledgements

With the completion of my diploma thesis, I would like to thank all the people that helped in preparing it. First, I would like to thank the supervising professor Mr. Dimitrios Soudris, for promoting my interests throughout my studies with the excellent work that he does in the Microprocessors and Digital Systems Lab, for trusting me with the thesis assignment and for his support throughout our collaboration, as well as PhD candidate Dimosthenis Masouros, who participated in the supervision, was always available to answer questions and address practical issues and contributed with his apt observations and suggestions. Moreover, I would like to thank Mr. Georgios Stamatakos, research professor at the Institute of Communication and Computer Systems, who shared with me his knowledge and enthusiasm for in silico medicine, who included me in the impeccable work he does in this field and who participated in the thesis supervision with creative ideas and suggestions, as well as his collaborators in the In Silico Oncology and In Silico Medicine Group, Eleni Georgiadi, Eleni Kolokotroni and Nikolaos Christodoulou, who were always available to give advice on practical matters concerning the Oncosimulator. I would also like to thank professor Norbert Graf and Marcel Meyerheim for their valuable contribution in preparing and sharing the patient medical data from Saarland University Hospital, without which it would be impossible to complete this work. Finally, I would like to thank my family and friends for always supporting me, believing in me and giving me courage.

# Abstract

The Oncosimulator is a top-down discrete entity-discrete event model for multiscale cancer modeling, that represents the tumor in silico as a 3D matrix of voxels and utilizes the «summarize and jump» strategy for the simulation of its evolution on the cellular and tissue level of biocomplexity. The Nephroblastoma Oncosimulator is a version of the Oncosimulator software that is specific to the Wilms' tumor, which is the most common kidney tumor in childhood. It was developed by members of the In Silico Oncology and In Silico Medicine Group, ICCS, ECE, National Technical University of Athens, under the lead of G. Stamatakos and the clinical advisorship of N. Graf, University of Saarland, from 2006 to 2019. The ultimate goal of the Oncosimulator and its neoplastic disease specific versions is to act as medical advisors for personalized medicine by creating a digital twin for the tumor and its normal tissue microenvironment. However, for that purpose the patient-specific medical data need to be translated in silico and, even though this is a trivial task when the imaging data and the size and shape of the tumor are considered, the values for the simulation input parameters that define the simulated biologic processes are not unambiguously derived. The purpose of the present work is to address this issue by exploring high performance computing methods for the performance enhancement of a single simulation and the efficient performance of a multitude of virtual patient simulations for each physical patient, with each virtual patient having a distinct value for the joint distribution of the simulation input parameters. The cluster of virtual patient simulations were executed for three already treated patients of distinct histologic profiles and corresponding risk groups, whose medical data were provided by the Saarland University Hospital, with each virtual patient being assigned a distinct value for the joint distribution of the simulation input parameters and the cell kill ratio parameter, which expresses the effects of the administered therapy, being explored for each virtual patient until the simulation outcomes matched the actual patient outcomes. Prior to the execution of the cluster of virtual patient executions that implement the in silico clinical adaptation of the $CKR$ parameter, the size of the voxel in the 3D tumor matrix was adjusted for the inputs that correspond to each dataset patient in the context of a data preprocessing step that addresses the simulation resolution and costs tradeoff problem, code optimizations were applied for the utilization of improved resources and the concurrent execution of computations when the data dependencies allow it and sensitivity analysis was performed for the verification of the preprocessed input data and the optimized source code via the comparison to predecessor verified simulation executions. The data preprocessing step introduces significant speedup in the simulation execution time (approximately $37 - 688\%$) and memory footprint reduction of the simulation execution (approximately $87{-}94\%$), while indirectly improving the speedup potential of the code optimization step according to Amdhal's law by rendering the simulation workload balance more fair and less intense on the parts that do not benefit from the improved resources. The source code optimization introduces a speedup of approximately $16 - 24\%$ by focusing on performing concurrent computations for the data dependencies free third simulation scan that concludes the simulation on the cellular level of biocomplexity. The final clinical adaptation step was perfomed for 20 and 200 virtual patients for each dataset patient, producing a distinct distribution, i.e. mean value and standard deviation, for the $CKR$ parameter for each distinct risk group of the patient dataset.

**Keywords:** Oncosimulator, Nephroblastoma Oncosimulator, Wilms' tumor, In silico medicine, Multiscale cancer modeling, Digital twin, High performance computing, Valgrind, OpenMP, Virtual patients, Clinical adaptation

# Εκτενής Περίληψη

## Εισαγωγή

### Τί είναι η πολυκλιμακωτή προσομοίωση της ασθένειας του καρκίνου;

Ο καρκίνος είναι μία νεοπλασματική ασθένεια, με την έννοια ότι συνοδεύεται από τον ανώμαλο και ανεξέλεγκτο πολλαπλασιασμό των κυττάρων, ο οποίος είναι γνωστός με την ονομασία νεόπλασμα. Το νεόπλασμα είναι το αποτέλεσμα της νεοπλασίας, δηλαδή της από κοινού δράσης βιολογικών διεργασιών που οδηγούν στον ανεξέλεγκτο πολλαπλασιασμό των κυττάρων. Τα νεοπλάσματα συνήθως δημιουργούν όγκους, οι οποίοι είναι είτε καλοήθεις είτε κακοήθεις. Στην περίπτωση του καρκίνου, οι όγκοι είναι κακοήθεις και υπάρχει κίνδυνος για μετάσταση, δηλαδή για τη δημιουργία όγκων και σε άλλα σημεία του οργανισμού πέρα από την εστία εμφάνισης του πρωτοπαθούς όγκου.

Οι παράγοντες που προκαλούν την εμφάνιση του καρκίνου είναι τόσο γενετικοί όσο και περιβαλλοντικοί και επιδρούν στους βιολογικούς μηχανισμούς του οργανισμού οδηγώντας στη νεοπλασία, στον ανεξέλεγκτο πολλαπλασιασμό των κυττάρων και στη δημιουργία όγκων. Σε μία πρώτη επιφανειακή προσέγγιση, το φαινόμενο του καρκίνου αφορά στα κύτταρα και τους ιστούς που πολλαπλασιάζονται ανεξέλεγκτα. Παρόλα αυτά, ο καρκίνος είναι ένα περίπλοκο φαινόμενο, το οποίο αφορά στη συνολική λειτουργία του οργανισμού, στα μακρομόρια και στα σήματα που μεταφέρονται σε αυτόν, καθώς και στο σύνολο των χαρακτηριστικών και δυνατοτήτων που αποκτούν οι όγκοι για το συνεχή και δυναμικό πολλαπλασιασμό τους, τα οποία είναι γνωστά στη βιβλιογραφία ως hallmarks of cancer.

Συμπερασματικά, ο καρκίνος είναι ένα περίπλοκο φαινόμενο και τα χαρακτηριστικά του, τα οποία οδηγούν σε δυναμική νεοπλασία, επιδρούν στους βιολογικούς μηχανισμούς του οργανισμού σε διαφορετικά επίπεδα βιοπολυπλοκότητας. Τα διαφορετικά αυτά επίπεδα βιοπολυπλοκότητας στα οποία εμφανίζεται το φαινόμενο του καρκίνου αφορούν τόσο στη χωρική όσο και στη χρονική κλίμακα. Κάθε επίπεδο της κλίμακας συντίθεται από μονάδες που ανήκουν στο χαμηλότερο επίπεδο. Για παράδειγμα, η χωρική κλίμακα ξεκινάει από τα άτομα, προχωράει στα μόρια, στα κύτταρα, στους ιστούς και τελικά στα όργανα. Ακόμα, η αύξηση του επιπέδου στη χωρική κλίμακα, συνοδεύεται κατά κανόνα από αύξηση του επιπέδου και στη χρονική κλίμακα, από $nsec$ και $msec$ σε δευτερόλεπτα, λεπτά, ώρες, μέρες, μήνες κ.ό.κ., καθώς ο ρυθμός με τον οποίο εξελίσσονται τα βιολογικά φαινόμενα μειώνεται. Για παράδειγμα, τα κύτταρα πολλαπλασιάζονται εκτελώντας τον κυτταρικό κύκλο, ο οποίος καταλήγει στη διαίρεση του κυττάρου σε δύο θυγατρικά κύτταρα και χρειάζεται κάποιες ώρες για να ολοκληρωθεί, ενώ ένας όγκος μπορεί να χρειάζεται μέρες ή και μήνες για να διπλασιαστεί σε μέγεθος, ανάλογα με τον τύπο και την ιστολογία του.

Ο ρόλος της πολυκλιμακωτής προσομοίωσης της ασθένειας του καρκίνου είναι ο επαρκής συνδυασμός των χωροχρονικών κλιμάκων και η κατάλληλη ενσωμάτωση των επιπέδων τους που χαρακτηρίζουν τα βιολογικά φαινόμενα που αφορούν στο φαινόμενο της ασθένειας του καρκίνου, με σκοπό την μαθηματική μοντελοποίηση του φαινομένου και τελικά την in silico προσομοίωσή του, δηλαδή την προσομοίωσή του στον υπολογιστή. Τα μοντέλα της πολυκλιμακωτής προσομοίωσης του καρκίνου διακρίνονται σε συνεχή και διακριτά με βάση τα μαθηματικά εργαλεία που χρησιμοποιούν για τη μοντελοποίηση, καθώς και σε bottom-up και top-down ανάλογα με το αν ξεκινούν την προσομοίωση από το χαμηλότερο ή το ψηλότερο επίπεδο της χωροχρονικής κλίμακας.

Τα συνεχή μοντέλα χρησιμοποιούν μεθόδους από τη μαθηματική ανάλυση, έτσι ώστε να μοντελοποιήσουν το βιολογικό φαινόμενο με ένα σύστημα από διαφορικές ή ολοκληρωτικές εξισώσεις, ενώ τα διακριτά μοντέλα εκμεταλλεύονται το γεγονός ότι οι βιολογικές οντότητες εμφανίζονται σε διακριτά κβάντα (π.χ. κύτταρα) και χρησιμοποιούν διακριτά εργαλεία και απευθείας αλγοριθμική λογική για τη μαθηματική μοντελοποίηση. Έτσι, τα διακριτά μοντέλα είναι καλύτερα προσαρμοσμένα στη διακριτή φύση που έχουν οι πράξεις στον υπολογιστή, ενώ τα συνεχή μοντέλα πρέπει να μεταφραστούν με αριθμητικές μεθόδους για την υλοποίηση της in silico προσομοίωσης.

Τα bottom-up μοντέλα χρησιμοποιούν την ήδη υπάρχουσα a priori γνώση που υπάρχει σχετικά με τα βιολογικά φαινόμενα που προσομοιώνονται, με σκοπό τη δημιουργία ενός πλήρους μοντέλου το οποίο αποτελείται από επιμέρους components, καθένα από τα οποία είναι υπεύθυνο για τη μοντελοποίηση μίας συγκεκριμένης και καλά ορισμένης διεργασίας. Έτσι, η πρσομοίωση σε ένα συγκεκριμένο επίπεδο βιοπολυπλοκότητας συντίθεται από τα χαμηλότερα επίπεδα της χωροχρονικής κλίμακας. Αντίθετα, τα top-down μοντέλα χρησιμοποιούν την ήδη υπάρχουσα γνώση μόνο σε κομβικά σημεία για τη θεμελίωσή τους, ενώ αξιοποιούν κατά βάση την κλινική εμπειρία. Για το σκοπό αυτό, δέχονται ως είσοδο ιατρικά δεδομένα, ξεκινούν την προσομοίωση από το ψηλότερο επίπεδο βιοπολυπλοκότητας και συχνά χρησιμοποιούν τη στρατηγική «summarize and jump», με βάση την οποία γίνεται ένα «άλμα» στο επόμενο επίπεδο της προσομοίωσης αφού πρώτα γίνει η «σύνοψη» στο τρέχον επίπεδο, για τις μεταβάσεις στη χωροχρονική κλίμακα βιοπολυπλοκότητας.

## Τί είναι ο Ογκοπροσομοιωτής Νεφροβλαστώματος;

Ο Ογκοπροσομοιωτής είναι ένα διακριτό top-down μοντέλο για πολυκλιμακωτή προσομοίωση της ασθένειας του καρκίνου. Πιο συγκεκριμένα, αξιοποιεί τα ιατρικά δεδομένα που απεικονίζουν τον όγκο ενός ασθενή που πάσχει από κάποια νεοπλασματική ασθένεια, έτσι ώστε να αντιστοιχίσει το φυσικό όγκο σε ένα τρισδιάστατο πλέγμα από voxels και να προσομοιώσει τη χρονική του εξέλιξη στον υπολογιστή σε μία σειρά από διακριτά χρονικά βήματα. Ο Ογκοπροσομοιωτής χαρακτηρίζεται ως discrete entity-discrete event μοντέλο, αφού απεικονίζει το φυσικό όγκο σε ένα τρισδιάστατο πλέγμα από διακριτά voxels και προσομοιώνει τη χρονική του εξέλιξη σε διακριτά βήματα. Ακόμα, αποτελεί ένα top-down μοντέλο, αφού αξιοποιεί την κλινική εμπειρία χρησιμοποιώντας τα ιατρικά δεδομένα του ασθενή, ενώ χρησιμοποιεί τη «summarize and jump» στρατηγική για την προσομοίωση στο κυτταρικό και ιστικό επίπεδο βιοπολυπλοκότητας.

Το νεφροβλάστωμα, πιο γνωστό ως όγκος του Wilms, αποτελεί τον πιο συνήθη πρωτοπαθή νεφρικό όγκο της παιδικής ηλικίας. Η εμφάνισή του οφείλεται κυρίως σε γενετικούς παράγοντες, όπως μεταλλάξεις στα γονίδια WT1, WT2, WTX, FWT-1. Έτσι, συχνά ο όγκος του Wilms εμφανίζεται μαζί με κάποιο γενετικό σύνδρομο, π.χ. WAGR ή Denys-Drash, ενώ η σποραδική εμφάνιση της νόσου είναι πιο σπάνια. Ακόμα, ο όγκος του Wilms προκαλείται από την ανώμαλη ανάπτυξη του μετανεφρογόνου βλαστήματος, το οποίο δέχεται επιθηλιακή, βλαστηματική και στρωματική διαφοροποίηση. Έτσι, ο προκύπτων όγκος μπορεί να έχει μικτή τριφασική ή διφασική ιστολογία, ή και μονοφασική ιστολογία. Η πιο επικίνδυνη από τις τρεις κατηγορίες ιστού είναι η βλαστηματική, ενώ μία εξαιρετικά δυσμενής ιστολογία είναι η αναπλαστική, η οποία χαρακτηρίζεται από διεύρυνση και υπερχρωματισμό των πυρήνων των κυττάρων. Ακόμα, η προσβολή και των δύο νεφρών χαρακτηρίζεται ως διμερής εμπλοκή, ενώ όταν εμφανίζονται εστίες όγκων και σε άλλα σημεία του σώματος πέρα των νεφρών η νόσος χαρακτηρίζεται ως μεταστατική. Η νόσος κατατάσσεται κατά τη διάγνωση σε ένα από πέντε στάδια, ανάλογα τις προσβαλλόμενες ανατομικές περιοχές, αν είναι μονομερής ή διμερής και αν είναι μεταστατική ή όχι.

Ο Ογκοπροσομοιωτής Νεφροβλαστώματος είναι η εκδοχή του Ογκοπροσομοιωτή που είναι εξειδικευμένη για την in silico απεικόνιση του όγκου του Wilms. Αναπτύχθηκε από μέλη της Ομάδας για την In Silico Ογκολογία και την In Silico Ιατρική των ΕΠΙΣΕΥ, ΣΗΜΜΥ, ΕΜΠ, υπό την επιστημονική καθοδήγηση του Γ. Σταματάκου και την κλινική εποπτεία του N. Graf, Πανεπιστήμιο του Saarland, από το 2006 έως το 2019. Πιο συγκεκριμένα, χρησιμοποιεί τα ιατρικά απεικονιστικά δεδομένα ενός συγκεκριμένου ασθενή, δημιουργεί ένα digital twin για τον όγκο στον υπολογιστή και προσομοιώνει τη χρονική του εξέλιξη in silico σε διακριτά χρονικά βήματα. Εφόσον αξιοποιεί τα ιατρικά δεδομένα ενός συγκεκριμένου ασθενή, μπορεί να προβλέψει την εξέλιξη της ασθένειας για αυτόν τον ασθενή και έτσι να προσφέρει τελικά υπηρεσίες εξατομικευμένης ιατρικής. Παρόλο που το νεφροβλάστωμα χαρακτηρίζεται από εν γένει ευνοϊκή πρόγνωση, η εξέλιξη της νόσου είναι πιο επικίνδυνη όταν η ιστολογία είναι δυσμενής και όσο το στάδιο της νόσου αυξάνεται. Ακόμα, η επιλογή του σχεδίου για την αντιμετώπιση της νόσου είναι κρίσιμη, όχι μόνο για την επιβίωση του ασθενή, αλλά και για τα συνολικά του κλινικά αποτελέσματα, την πιθανότητα υποτροπής και τις παρενέργειες, βραχυπρόθεσμες και μακροπρόθεσμες.

Έτσι, ο ρόλος του Ογκοπροσομοιωτή Νεφροβλαστώματος για την πρόβλεψη της εξέλιξης της νόσου για το συγκεκριμένο ασθενή και στο πλαίσιο διαφορετικών θεραπευτικών πλάνων είναι εξαιρετικά σημαντικός, έτσι ώστε να εκτιμάται το πλάνο που είναι πιο ωφέλιμο σε κάθε κλινική περίπτωση χωριστά. Για παράδειγμα, ένα απλό κλινικό ερώτημα που θα μπορούσε να απαντήσει ο Ογκοπροσομοιωτής Νεφροβλαστώματος είναι αν είναι χρήσιμο για το συγκεκριμένο ασθενή να δεχθεί προεγχειρητική χημειοθεραπεία πριν την επέμβαση αφαίρεσης του όγκου. Η προεγχειρητική χημειοθεραπεία είναι όπως ορίζεται από το καθιερωμένο SIOP πρωτόκολλο και περιλαμβάνει τη χορήγηση δύο χημειοθεραπευτικών φαρμάκων, της Ακτινομυκίνης και της Βινκριστίνης, σε ένα διάστημα τεσσάρων εβδομάδων. Αν η προεγχειρητική χημειοθεραπεία προκαλεί επαρκή σμίκρυνση του όγκου, τότε έχει νόημα η εφαρμογή της, αλλιώς μόνο προκαλεί παρενέργειες και καθυστερεί την πρόοδο της θεραπείας.

### Η ανάγκη για χρήση πολλαπλών εικονικών ασθενών ανά φυσικό ασθενή, ο ρόλος του high performance computing και ο σκοπός αυτής της εργασίας

Συμπερασματικά, ο Ογκοπροσομοιωτής δημιουργεί ένα digital twin για τον όγκο και τον κανονικό ιστό που τον περιβάλλει, έτσι ώστε να μπορεί να προσομοιωθεί in silico η χρονική του εξέλιξη, στο πλαίσιο διαφόρων θεραπευτικών πλάνων ή ακόμα και για ελεύθερη ανάπτυξη. Για το σκοπό αυτό, αξιοποιούνται τα ιατρικά δεδομένα του συγκεκριμένου ασθενή που πάσχει από κάποια νεοπλασματική νόσο και τελικά ο Ογκοπροσομοιωτής μπορεί να χρησιμοποιηθεί ως εργαλείο για εξατομικευμένη ιατρική και να λειτουργεί ως σύμβουλος που προτείνει την καλύτερη θεραπευτική αντιμετώπιση για τον εκάστοτε ασθενή.

Για να είναι όμως αυτό δυνατό, πρέπει να γίνει αποτελεσματική και ακριβής μετάφραση των ιατρικών δεδομένων του ασθενή στον υπολογιστή και σε μορφή που μπορεί να αξιοποιηθεί από τον Ογκοπροσομοιωτή. Αυτή η διαδικασία είναι εύκολη όσον αφορά στη μορφολογία του όγκου και στα απεικονιστικά δεδομένα, τα οποία καθορίζουν την αρχικοποίηση του τρισδιάστατου πλέγματος που παριστάνει in silico το φυσικό όγκο και τον κανονικό ιστό που τον περιβάλλει. Ωστόσο, η πρόοδος της προσομοίωσης εξαρτάται από τις βιολογικές διεργασίες που προσομοιώνονται, οι οποίες εξελίσσονται με βάση τις τιμές των παραμέτρων εισόδου του μοντέλου, οι οποίες συνήθως δεν καθορίζονται άμεσα από τα ιατρικά δεδομένα. Για παράδειγμα, οι χρονικές στιγμές χορήγησης των φαρμάκων του προσομοιούμενου χημειοθεραπευτικού πλάνου καθορίζονται εύκολα, αλλά το ποσοστό των κυττάρων που χτυπάει η θεραπεία, το ποσοστό των κυττάρων που εκτελούν συμμετρική διαίρεση και το ποσοστό των κυττάρων που εισέρχονται στην κοιμώμενη φάση εκτός του κυτταρικού κύκλου, τα οποία εκφράζονται από τις παραμέτρους $CKR$, $P_{sym}$ και $P_{sleep}$ αντίστοιχα, δεν καθορίζονται μονοσήμαντα από τα ιατρικά δεδομένα του ασθενή.

Η ανάθεση τιμών στις παραμέτρους του μοντέλου εισάγει, λοιπόν, μια αβεβαιότητα όσον αφορά στην εκτέλεση προσομοιώσεων που παριστάνουν και προβλέπουν κατάλληλα την κλινική πορεία του εκάστοτε ασθενή. Για να αντιμετωπιστεί αυτό το πρόβλημα, είναι αναγκαία η χρήση πολλαπλών εικονικών ασθενών ανά φυσικό ασθενή. Πιο συγκεκριμένα, εκτελούνται πολλαπλές προσομοιώσεις για κάθε φυσικό ασθενή, με κάθε προσομοίωση να αντιστοιχεί σε διακριτή τιμή για την από κοινού κατανομή των παραμέτρων εισόδου του μοντέλου που δεν καθορίζονται μονοσήμαντα από τα ιατρικά δεδομένα του ασθενή. Έτσι, δημιουργούνται πολλοί εικονικοί ασθενείς για κάθε φυσικό ασθενή, καθένας από τους οποίους εκπροσωπεί μία διαφορετική εκδοχή του βιολογικού συστήματος και εν γένει διαφορετικές προβλέψεις και αποτελέσματα. Αξιολογώνατς τα αποτελέσματα των εικονικών ασθενών από κοινού, εκτελώντας για παράδειγμα κατάλληλη στατιστική ανάλυση πάνω σε αυτά, επιτυγχάνεται μία αντιπροσωπευτική in silico αναπαράσταση του φυσικού ασθενή και της κλινικής του εικόνας και πορείας. Ακόμα, εφόσον υπάρχει στόχος για χρήση του Ογκοπροσομοιωτή στην κλινική πρακτική, πρέπει το σύστημα των προσομοιώσεων εικονικών ασθενών να ικανοποιεί κάποια αυστηρά κριτήρια απόδοσης, έτσι ώστε να παρέχονται τα αποτελέσματα σε πραγματικό χρόνο, γεγονός που υπογραμμίζει το ρόλο του high performance computing.

Σε αυτή την εργασία, έγινε εξερεύνηση high performance computing μεθόδων για τον Ογκοπροσομοιωτή Νεφροβλαστώματος, για τη βελτίωση της απόδοσης της εκτέλεσης μίας μεμονωμένης προσομοίωσης, αλλά και για την αποδοτική εκτέλεση πολλαπλών προσομοιώσεων εικονικών ασθενών ανά φυσικό ασθενή. Έπειτα, η αρχιτεκτονική εκτέλεσης μαζικών προσομοιώσεων αξιοποιήθηκε για την κλινική προσαρμογή των παραμέτρων εισόδου του μοντέλου, έτσι ώστε να βρεθούν οι τιμές παραμέτρων που παριστάνουν πιο κατάλληλα την κλινική εικόνα των φυσικών ασθενών στον υπολογιστή.

Πιο συγκεκριμένα, το σύνολο δεδομένων αποτελείται από τρεις ασθενείς, οι οποίοι έχουν ήδη δεχθεί χημειοθεραπεία και χειρουργική αφαίρεση του όγκου και τα ιατρικά δεδομένα των οποιών, που περιλαμβάνουν πληροφορίες σχετικά με το προεγχειρητικό χημειοθεραπευτικό πλάνο που εφαρμόστηκε, απεικονιστικά δεδομένα πριν και μετά την εφαρμογή του και ιστοπαθολογικά δεδομένα όπως προκύπτουν μετά από τη χειρουργική αφαίρεση και βιοψία του όγκου, προσφέρονται από το πανεπιστημιακό νοσοκομείο του Saarland. Ακόμα, καθένας από τους τρεις ασθενείς του συνόλου δεδομένων εκπροσωπεί ένα διακριτό ιστοπαθολογικό προφίλ και αντίστοιχη ομάδα κινδύνου και η κλινική προσαρμογή έγινε για την παράμετρο CKR, η οποία καθορίζει το ποσοστό των κυττάρων που χτυπάει η θεραπεία και εκφράζει έτσι την αποτελεσματικότητά της, με στόχο τα αποτελέσματα της προσομοίωσης να προσεγγίζουν επαρκώς τα πραγματικά κλινικά αποτελέσματα. Έτσι, εκτελώντας κατάλληλη στατιστική ανάλυση στο σύνολο των προσομοιώσεων εικονικών ασθενών της κλινικής προσαρμογής, προκύπτει η κατανομή πιθανότητας της παραμέτρου CKR για κάθε μία από τις τρεις ομάδες κινδύνου που εκπροσωπούνται στο σύνολο δεδομένων, ως ζευγάρι μέσης τιμής και τυπικής απόκλισης.

## Θεωρητικό Υπόβαθρο

### Ο κυτταρικός κύκλος ως το βιολογικό φαινόμενο που προσομοιώνεται στο κυτταρικό επίπεδο

Κάθε ευκαρυωτικό κύτταρο εκτελεί τον κυτταρικό κύκλο για τον πολλαπλασιασμό του. Ο κυτταρικός κύκλος αποτελείται από τέσσερις φάσεις, τις $G_1$, $S$, $G_2$ και $M$. Οι φάσεις $G_1$ και $G_2$, γνωστές ως gap phases, χαρακτηρίζονται από έντονη βιοχημική δραστηριότητα και προετοιμάζουν το κύτταρο για τις φάσεις $S$ και $M$ αντίστοιχα. Η φάση $S$ είναι η φάση σύνθεσης του DNA, όπου ουσιαστικά γίνεται η αντιγραφή των χρωμοσωμάτων του, με αποτέλεσμα κάθε χρωμόσωμα να αποτελείται από δύο αδερφές χρωματίδες. Η φάση $M$ είναι η φάση της μίτωσης, κατά την οποία γίνεται η πυρηνική διαίρεση, ο διαχωρισμός των χρωμοσωμάτων του κυττάρου σε δύο πανομοιότυπα σύνολα και τελικά η διαίρεση του κυττάρου σε δύο πανομοιότυπα κύτταρα που μοιράζονται εξίσου τα συστατικά του γονικού κυττάρου. Έτσι, ολοκληρώνοντας τον κυτταρικό κύκλο, το ευκαρυωτικό κύτταρο πολλαπλασιάζεται μέσω της διαίρεσής του σε δύο πανομοιότυπα θυγατρικά κύτταρα.

Ο κυτταρικός κύκλος είναι μία σειρά από αυστηρά ελεγχόμενες βιολογικές διεργασίες και γεγονότα, τα οποία ρυθμίζονται με βάση ένα δυναμικό δίκτυο από μακρομόρια και σήματα που μεταφέρονται στον οργανισμό. Για παράδειγμα, η μετάβαση από τη μία φάση του κυτταρικού κύκλου στην επόμενη γίνεται με βάση τη δημιουργία δεσμών από ειδικές πρωτεΐνες που ονομάζονται cyclins, οι οποίες ενεργοποιούν ειδικά ένζυμα που ονομάζονται CDKs (cyclin-dependent kinases). Αυτός ο αυστηρός έλεγχος της σωστής διεξαγωγής του κυτταρικού κύκλου είναι εξαιρετικά σημαντικός, καθώς η διαταραχή του κυτταρικού κύκλου μπορεί να προκαλέσει παθολογικές καταστάσεις, όπως είναι η νεοπλασία, ο ανεξέλεγκτος κυτταρικός πολλαπλασιασμός και η εμφάνιση της ασθένειας του καρκίνου.

Για αυτό το λόγο, υπάρχουν κάποια checkpoints, ένα σύνολο κριτηρίων που αφορούν σε συγκεκριμένα σημεία της εξέλιξης του κυτταρικού κύκλου και βεβαιώνουν ότι αυτή συντονίζεται σωστά. Ένα πολύ σημαντικό checkpoint αφορά στη φάση $G_1$ στην αρχή του κυτταρικού κύκλου και ελέγχει ότι υπάρχουν οι κατάλληλες συνθήκες για την ανάπτυξη του κυττάρου, π.χ. θρεπτικά συστατικά και κατάλληλοι παράγοντες ανάπτυξης. Αν πληρούνται αυτές οι συνθήκες, το κύτταρο συνεχίζει στη φάση $S$ και δεσμεύεται να πολλαπλασιαστεί ολοκληρώνοντας τον κυτταρικό κύκλο. Αλλιώς, εισέρχεται στην κοιμώμενη φάση $G_0$ εκτός του κυτταρικού κύκλου, όπου διατηρεί το ρόλο του αλλά δεν πολλαπλασιάζεται, ενώ ενδέχεται να εισέλθει ξανά στον κυτταρικό κύκλο από τη φάση $G_1$, εφόσον αποκατασταθούν οι απαραίτητες συνθήκες και παράγοντες ανάπτυξης.

Ο Ογκοπροσομοιωτής εκτελεί την προσομοίωση στο κυτταρικό επίπεδο με βάση το κυτταροκινητικό διάγραμμα, το οποίο περιλαμβάνει ένα σύνολο από πεπερασμένες καταστάσεις, οι οποίες εκφράζουν βιολογικές καταστάσεις/διεργασίες του κυττάρου. Πιο συγκεκριμένα, υλοποιεί τις μεταβάσεις μεταξύ αυτών των καταστάσεων με βάση τις τιμές των παραμέτρων εισόδου του μοντέλου και με αυτό τον τρόπο προσομοιώνει την εξέλιξη των βιολογικών φαινομένων στο κυτταρικό επίπεδο. Οι καταστάσεις του κυτταροκινητικού διαγράμματος αφορούν στις φάσεις εντός και εκτός του κυτταρικού κύκλου, στην αυθόρμητη απόπτωση ή νέκρωση που ενδέχεται να υποστεί κάποιο κύτταρο, στην εξολόθρευση των κυττάρων από τη χορηγούμενη θεραπεία, στην ασύμμετρη διαίρεση που προκαλεί τη δημιουργία κυττάρων που εκτελούν τον κυτταρικό κύκλο και πολλαπλασιάζονται πεπερασμένο πλήθος φορών

(LIMP cells), κ.λπ. Οι παράμετροι του μοντέλου καθορίζουν την υλοποίηση των μεταβάσεων του διαγράμματος, όπου αυτές δε γίνονται με ντετερμινιστικό τρόπο.

### Η ορολογία του υπερπίνακα και ο αλγόριθμος του Ογκοπροσομοιωτή

Όπως αναφέρθηκε προηγουμένως, ο Ογκοπροσομοιωτής αντιστοιχίζει το φυσικό όγκο και τον κανονικό ιστό που τον περιβάλλει σε ένα τρισδιάστατο πλέγμα από voxels. Κάθε voxel του τρισδιάστατου πλέγματος καταλαμβάνει μία διακριτή περιοχή στο τρισδιάστατο χώρο, για αυτό τα voxels ονομάζονται αλλιώς και γεωμετρικά κύτταρα. Κάθε γεωμετρικό κύτταρο περικλείει ένα πλήθος από βιολογικά κύτταρα. Όσο μεγαλύτερο είναι το μέγεθος του γεωμετρικού κυττάρου, τόσο μεγαλύτερη είναι και η γεωμετρική περιοχή που καταλαμβάνει και συνεπώς και το πλήθος των βιολογικών κυττάρων που περιέχει, υποθέτοντας ότι η πυκνότητα των βιολογικών κυττάρων στο χώρο είναι σταθερή. Μέσα σε κάθε γεωμετρικό κύτταρο, τα βιολογικά κύτταρα είναι περαιτέρω οργανωμένα σε κλάσεις ισοδυναμίας, με κάθε κλάση να αντιστοιχεί σε μία από τις καταστάσεις του κυτταροκινητικού διαγράμματος.

Έτσι, οι μεταβάσεις του κυτταροκινητικού διαγράμματος που υλοποιεί ο Ογκοπροσομοιωτής για την προσομοίωση στο κυτταρικό επίπεδο, εκτελούνται από κλάσεις ισοδυναμίας κυττάρων και όχι από μεμονωμένα βιολογικά κύτταρα. Η μοντελοποίηση της παράστασης του όγκου ως ένα τρισδιάστατο πλέγμα από γεωμετρικά κύτταρα, κάθε ένα από τα οποία περιέχει ένα πλήθος από βιολογικά κύτταρα τα οποία είναι οργανωμένα σε κλάσεις ισοδυναμίας, γίνεται με τη βοήθεια της ορολογίας του υπερπίνακα. Η αναφορά σε κάθε κλάση ισοδυναμίας, δηλαδή στη στοιχειώδη μονάδα του συστήματος, γίνεται με βάση το γεωμετρικό κύτταρο στο οποίο αυτή ανήκει, δηλαδή τις συντεταγμένες $x$, $y$, $z$, με βάση τη χρονική στιγμή $t$, καθώς και την φάση/κατάσταση του κυτταροκινητικού διαγράμματος $p$ που εκπροσωπεί.

Ο Ογκοπροσομοιωτή προσομοιώνει τη χρονική εξέλιξη του όγκου σε διακριτά χρονικά βήματα. Για αυτό το σκοπό, εφαρμόζει μία σειρά από διαδοχικά σκαναρίσματα στο τρισδιάστατο πλέγμα που παριστάνει τον όγκο in silico. Αυτά τα σκαναρίσματα λειτουργούν ως τελεστές που δρουν πάνω στο τρισδιάστατο πίνακα από voxels και έτσι τον μετασχηματίζουν. Το πρώτο σκανάρισμα του αλγορίθμου αφορά στην αρχικοποίηση του τρισδιάστατου πλέγματος με βάση τα απεικονιστικά δεδομένα του εκάστοτε ασθενή και εκτελείται μόνο μία φορά στην αρχή της προσομοίωσης. Το δεύτερο σκανάρισμα καθορίζει τα βιολογικά κύτταρα που χτυπάει η θεραπεία με βάση την παράμετρο CKR, ενώ το τρίτο σκανάρισμα υλοποιεί τις υπόλοιπες μεταβάσεις του κυτταροκινητικού διαγράμματος με βάση τις παραμέτρους εισόδου του μοντέλου. Έτσι, το δεύτερο και τρίτο σκανάρισμα εκτελούνται σε κάθε χρονικό βήμα του αλγορίθμου για την υλοποίηση της προσομοίωσης στο κυτταρικό επίπεδο.

Το τέταρτο σκανάρισμα φροντίζει για την απαλλαγή των γεωμετρικών κυττάρων από περίσσεια βιολογικών κυττάρων και για την αύξηση του όγκου, ενώ το πέμπτο σκανάρισμα φροντίζει για την απαλοιφή των γεωμετρικών κυττάρων που περιέχουν πολύ λίγα βιολογικά κύτταρα και για τη σμίκρυνση του όγκου. Το έκτο σκανάρισμα εφαρμόζεται για τη διατήρηση της ομοιογένειας του συμπαγούς όγκου μετά από τις παρεμβάσεις των δύο προηγούμενων σκαναρισμάτων. Έτσι, το τέταρτο, πέμπτο και έκτο σκανάρισμα εκτελούνται σε κάθε χρονικό βήμα του αλγορίθμου για την υλοποίηση της προσομοίωσης στο ιστικό επίπεδο, όπου ο όγκος ενδέχεται να αυξηθεί ή να σμικρυνθεί παραμένοντας όμως πάντα συμπαγής. Άρα, σε κάθε χρονικό βήμα ο Ογκοπροσομοιωτής εκτελεί μία «σύνοψη» στο κυτταρικό επίπεδο και ένα «άλμα» στο ιστικό επίπεδο, εφαρμόζοντας έτσι τη στρατηγική «summarize and jump». Αυτή η διαδικασία επαναλαμβάνεται μέχρι την ολοκλήρωση της προσομοίωσης, όπου εφαρμόζεται το τελικό έβδομο σκανάρισμα, για την τελική απομάκρυνση των γεωμετρικών κυττάρων που περιέχουν πολύ λίγα βιολογικά κύτταρα.

## Μεθοδολογία

### Data preprocessing

Το πρώτο βήμα που υλοποιήθηκε σε αυτή την εργασία αφορά στην προετοιμασία της εισόδου, με στόχο τη σωστή και αποτελεσματική εκτέλεση των προσομοιώσεων, για κάθε φυσικό ασθενή του συνόλου δεδομένων. Για αυτό το σκοπό, έγινε η προεπεξεργασία των ιατρικών δεδομένων που παρέχονται από το πανεπιστημιακό νοσοκομείο του Saarland, έτσι ώστε να αποκτήσουν την κατάλληλη μορφή για να χρησιμοποιηθούν ως είσοδος του Ογκοπροσομοιωτή Νεφροβλαστώματος, να γίνει αποδοτικά η εκτέλεση της προσομοίωσης και να ολοκληρωθεί σωστά παράγοντας τα επιθυμητά αποτελέσματα. Το βήμα της προεπεξεργασίας δεδομένων εξασφαλίζει ότι η εκτέλεση της προσομοίωσης θα ολοκληρωθεί σωστά, αλλά και ότι θα γίνει με το μικρότερο δυνατό κόστος. Έτσι, είναι σημαντικό να εκτελείται πριν ξεκινήσει η αναζήτηση τρόπων βελτίωσης της απόδοσης του μοντέλου με αλλαγές στον πηγαίο κώδικα και στον τρόπο με τον οποίο εκτελούνται οι υπολογισμοί.

Ο Ογκοπροσομοιωτής Νεφροβλαστώματος δέχεται ως είσοδο τα απεικονιστικά δεδομένα που περιγράφουν το μέγεθος, το σχήμα και τη συνολική μορφολογία του όγκου, μέσω δύο αρχείων εισόδου. Το πρώτο αρχείο εισόδου, με επέκταση .raw, αποτελεί ένα byte stream που περιέχει την ακατέργαστη πληροφορία σχετικά με την κατηγορία ιστού στην οποία ανήκει κάθε γεωμετρικό κύτταρο του τρισδιάστατου πλέγματος, ενώ το δεύτερο αρχείο εισόδου, με επέκταση .mhd, αποτελεί ένα metaheader αρχείο που συνοδεύει το .raw αρχείο και περιγράφει τα χαρακτηριστικά του τρισδιάστατου πλέγματος, όπως τις διαστάσεις του και το μέγεθος του γεωμετρικού κυττάρου. Πιο συγκεκριμένα, αν το μέγεθος της κάθε διάστασης ορίζεται ως $x\_dim$, $y\_dim$ και $z\_dim$ στο .mhd αρχείο, τότε το μέγεθος του .raw αρχείου σε bytes είναι $x\_dim \cdot y\_dim \cdot z\_dim$ και η ιστολογική κατηγορία στην οποία ανήκει το γεωμετρικό κύτταρο με συντεταγμένες $(i, j, k)$ καθορίζεται από το byte στη θέση $i + x\_dim \cdot j + x\_dim \cdot y\_dim \cdot k$ (τιμή 0x00 αντιστοιχεί σε κανονικό ιστό, ενώ τιμή 0xFF αντιστοιχεί σε ιστό του όγκου).

Ακόμα, στο .mhd αρχείο καθορίζεται το μέγεθος του γεωμετρικού κυττάρου του τρισδιάστατου πλέγματος που παριστάνει in silico το φυσικό όγκο και τον κανονικό ιστό που τον περιβάλλει. Όσο μεγαλώνει το μέγεθος του γεωμετρικού κυττάρου, αυξάνεται η γεωμετρική περιοχή του αυτό καλύπτει στον τρισδιάστατο χώρο και συνεπώς μειώνεται το πλήθος των γεωμετρικών κυττάρων που χρησιμοποιείται ανά διάσταση για να καλύψει τη φυσική περιοχή που αντιστοιχεί στον όγκο και στον κανονικό ιστό που τον περιβάλλει. Έτσι, μειώνεται το μέγεθος εισόδου, δηλαδή το μέγεθος του τρισδιάστατου πλέγματος του όγκου, με αποτέλεσμα να μειώνεται και το κόστος της προσομοίωσης, τόσο όσον αφορά στις απαιτήσεις για χώρο στη μνήμη όσο και αναφορικά με το συνολικό χρόνο εκτέλεσης. Ταυτόχρονα, όμως, όσο το μέγεθος του γεωμετρικού κυττάρου και η περιοχή που καλύπτει στο χώρο αυξάνονται, αυξάνεται και το πλήθος των βιολογικών κυττάρων που αυτό περικλείει, με αποτέλεσμα να περιλαμβάνονται περισσότερα βιολογικά κύτταρα σε κάθε κλάση ισοδυναμίας του γεωμετρικού κυττάρου, να εκτελούνται οι μεταβάσεις του κυτταροκινητικού διαγράμματος από πολλά βιολογικά κύτταρα από κοινού και τελικά να μειώνεται η ανάλυση της προσομοίωσης. Έτσι, το μέγεθος του γεωμτερικού κυττάρου συνδέεται με ένα tradeoff ανάμεσα στο κόστος και στην ανάλυση της προσομοίωσης.

Στο πλαίσιο της προεπεξεργασίας των δεδομένων εισόδου του Ογκοπροσομοιωτή Νεφροβλαστώματος, υλοποιήθηκε ένας αλγόριθμος για τη ρύθμιση του μεγέθους του γεωμετρικού κυττάρου του τρισδιάστατου πλέγματος που παριστάνει τον όγκο στον υπολογιστή, με σκοπό τη βέλτιστη αντιμετώπιση του tradeoff κόστους και ανάλυσης της προσομοίωσης για κάθε φυσικό ασθενή του συνόλου δεδομένων. Πιο συγκεκριμένα, το μέγεθος του γεωμετρικού κυττάρου ρυθμίζεται με βάση ένα scale factor, το οποίο είναι αρχικά ίσο με 1 και αυξάνεται συνεχώς, με αποτέλεσμα να αυξάνεται σταδιακά και το μέγεθος του γεωμετρικού κυττάρου. Έτσι, εκτελούνται διαδοχικές προσομοιώσεις με το μέγεθος του γεωμετρικού κυττάρου να αυξάνεται, γεγονός που έχει ως αποτέλεσμα την ταυτόχρονη μείωση του κόστους και της ανάλυσης της προσομοίωσης. Αυτή η διαδικασία συνεχίζεται μέχρι το σφάλμα που εισάγει η έκπτωση στην ανάλυση στα αποτελέσματα της προσομοίωσης σε σχέση με αυτά που καταγράφονται για την αρχική ανάλυση με scale factor ίσο με 1, το οποίο υπολογίζεται με βάση το προσομοιούμενο ποσοστό μείωσης του όγκου, να ξεπεράσει ένα συγκεκριμένο κατώφλι. Μετά από την εκτέλεση του αλγορίθμου για κάθε φυσικό ασθενή του συνόλου δεδομένων, γίνεται αξιολόγηση των αποτελεσμάτων, έτσι ώστε να καθοριστεί η τιμή του scale factor που μειώνει επαρκώς το χρόνο εκτέλεσης χωρίς να εισάγει αισθητό σφάλμα στα αποτελέσματα.

### Profiling & Code optimization

Εφόσον έχει ολοκληρωθεί το στάδιο της προεπεξεργασίας των δεδομένων εισόδου του Ογκοπροσομοιωτή και έχουν καθοριστεί τα μεγέθη του γεωμετρικού κυττάρου που αντιμετωπίζουν βέλτιστα το tradeoff μεταξύ κόστους και ανάλυσης της προσομοίωσης για κάθε φυσικό ασθενή του συνόλου δεδομένων, μπορούν να εξερευνηθούν μέθοδοι για τη βελτίωση της απόδοσης της εκτέλεσης της προσομοίωσης που αφορούν στον τρόπο με τον οποίο γίνονται οι υπολογισμοί και στον πηγαίο κώδικα του μοντέλου. Για αυτό το σκοπό, έγινε profiling για τον καθορισμό του κόστους της εκτέλεσης της προσομοίωσης, τόσο όσον αφορά στις απαιτήσεις σε χώρο στη μνήμη όσο και αναφορικά με το χρόνο εκτέλεσης, καθώς και για την κατανομή του κόστους στα σημεία του πηγαίου κώδικα που συμμετέχουν σε αυτό. Έπειτα, με βάση τις πληροφορίες που προέκυψαν από αυτό το profiling, έγινε η βελτιστοποίηση του μοντέλου, έτσι ώστε να εκτελούνται παράλληλα οι υπολογισμοί στα κοστοβόρα σημεία του πηγαίου κώδικα, όπου οι εξαρτήσεις δεδομένων το επιτρέπουν.

Το profiling έγινε με τη βοήθεια του valgrind framework, το οποίο αποτελείται από μία σειρά εργαλείων που συνοδεύουν την εκτέλεση, παρακολουθούν και καταγράφουν την εξέλιξή της και προσφέρουν κάποιες εκτιμήσεις σχετικά με αυτή. Πιο συγκεκριμένα, για την εκτίμηση των απαιτήσεων της προσομοίωσης σε χώρο στη μνήμη χρησιμοποιήθηκε το εργαλείο massif, το οποίο παρακολουθεί τη χρήση του σωρού κατά τη διάρκεια της εκτέλεσης. Ο Ογκοπροσομοιωτής Νεφροβλαστώματος δεσμεύει τη μνήμη που χρειάζεται για την παράσταση του φυσικού όγκου σε ένα τρισδιάστατο πλέγμα από voxels κατά το στάδιο της αρχικοποίησης, το μέγεθος του τρισδιάστατου πίνακα δεν αλλάζει κατά τη διάρκεια της προσομοίωσης και ο χώρος που καταλαμβάνει στη μνήμη αποδεσμεύεται με την ολοκλήρωση της προσομοίωσης. Έτσι, το εργαλείο massif δείχνει την αξιοποίηση του σωρού κατά τη διάρκεια της προσομοίωσης, η οποία είναι κατά βάση σταθερή, προσφέροντας μία εκτίμηση για τις απαιτήσεις της προσομοίωσης σε χώρο στη μνήμη, οι οποίες αλλάζουν ανάλογα με το μέγεθος της εισόδου.

Ακόμα, για τον καθορισμό των πιο κοστοβόρων τμημάτων της εκτέλεσης χρησιμοποιήθηκε το callgrind εργαλείο, το οποίο παρακολουθεί την εκτέλεση της προσομοίωσης και παράγει το callgraph που περιγράφει τις κλήσεις μεταξύ των συναρτήσεων του πηγαίου κώδικα και το ποσοστό με το οποίο η κάθε συνάρτηση συμμετέχει στο συνολικό κόστος της εκτέλεσης, όπως αυτό υπολογίζεται με βάση το CPU cycles count. Το εργαλείο callgrind εκτελέστηκε με ενεργοποιημενη την επιλογή για προσομοίωση της ιεραρχίας κρυφής μνήμης. Αυτό σημαίνει ότι κατά το profiling προσομοιώνεται το πρώτο και το τελευταίο επίπεδο της κρυφής μνήμης, μπορεί να εκτιμηθεί η αλληλεπίδραση της εκτέλεσης με την κρυφή μνήμη και την κύρια μνήμη, καθώς και το κατά πόσο το μοτίβο με το οποίο γίνονται οι προσβάσεις στη μνήμη αυξάνει το κόστος της εκτέλεσης. Το CPU cycles count υπολογίζεται υποθέτοντας ότι η εκτέλεση κάθε εντονής απαιτεί έναν κύκλο, η πρόσβαση στο τελευταίο επίπεδο της κρυφής μνήμης απαιτεί 10 κύκλους, ενώ η πρόσβαση στην κύρια μνήμη απαιτεί 100 κύκλους.

Μετά τον καθορισμό των πιο κοστοβόρων για την εκτέλεση τμημάτων που πηγαίου κώδικα, μπορούν να σχεδιαστούν και να υλοποιηθούν μέθοδοι για τη μείωση του κόστους και τη βελτίωση της απόδοδσης της εκτέλεσης. Αυτό έγινε χρησιμοποιώντας το OpenMP API, το οποίο είναι ένα μοντέλο για χρήση πολλαπλών threads σε αρχιτεκτονικές κοινής μνήμης. Πιο συγκεκριμένα, τα πολλαπλά threads χρησιμοποιήθηκαν για την ταυτόχρονη εκτέλεση των υπολογισμών του αλγορίθμου, όπου οι εξαρτήσεις δεδομένων το επιτρέπουν. Στην περίπτωση του Ογκοπροσομοιωτή, οι εξαρτήσεις δεδομένων επιτρέπουν την ταυτόχρονη εκτέλεση υπολογισμών μόνο για το τρίτο σκανάρισμα του αλγορίθμου που ολοκληρώνει την προσομοίωση στο κυτταρικό επίπεδο για κάθε χρονικό βήμα της προσομοίωσης. Στο τρίτο σκανάρισμα της προσομοίωσης υλοποιούνται οι μεταβάσεις του κυτταροκινητικού διαγράμματος, οι οποίες εκτελούνται από κλάσεις ισοδυναμίας κυττάρων μέσα σε κάθε voxel του τρισδιάστατου πλέγματος του όγκου. Έτσι, οι υπολογισμοί που γίνονται για κάθε voxel είναι ανεξάρτητοι και μπορούν να εκτελεστούν παράλληλα. Αντίθετα, το τέταρτο, πέμπτο και έκτο σκανάρισμα, που ολοκληρώνουν την προσομοίωση στο ιστικό επίπεδο για κάθε χρονικό βήμα της προσομοίωσης, υλοποιούν τη ροή βιολογικών κυττάρων μεταξύ γειτονικών voxel για την αύξηση ή σμίκρυνση του όγκου, με αποτέλεσμα να εκτελούν υπολογισμούς που περιλαμβάνουν πολλές εξαρτήσεις και race conditions.

Η προσομοιώσεις εκτελέστηκαν σε ένα linux server που χρησιμοποιεί 2 επεξεργαστές, καθένας από τους οποίους αποτελείται από 12 φυσικούς πυρήνες. Ακόμα, κάθε πυρήνας τρέχει 2 threads, με αποτέλεσμα το σύνολο των διαθέσιμων λογικών πυρήνων να είναι ίσο με 48. Ακόμα, κάθε φυσικός πυρήνας (24 στο σύνολο) έχει ξεχωριστά τα 2 πρώτα επίπεδα της ιεραρχίας κρυφής μνήμης ($L1d$ και $L1i$ $32K$ η κάθε μία και $L2$ $256K$), ενώ το τρίτο και τελευταίο επίπεδο ($L3$ $30720K$) μοιράζεται μεταξύ των φυσικών πυρήνων που ανήκουν στον ίδιο επεξεργαστή (2 στο σύνολο). Η μετρική που χρησιμοποιήθηκε για την αξιολόγηση της βελτίωσης της απόδοσης με τις αλλαγές στον πηγαίο κώδικα για παράλληλη εκτέλεση υπολογισμών με χρήση πολλαπλών threads είναι το σχετικό speedup, το οποίο υπολογίζεται ως το κλάσμα του αρχικού χρόνου εκτέλεσης δία το χρόνο εκτέλεσης μετά τη βελτιστοποίηση του πηγαίου κώδικα. Το μέγιστο εφικτό speedup περιορίζεται με βάση το νόμο του Amdahl, ο οποίος λέει ότι, υποθέτοντας ότι $p$ είναι το τμήμα του φόρτου εργασίας που αξιοποιεί τους βελτιωμένους πόρους, εν προκειμένω τα πολλαπλά threads, και $s$ το σχετικό speedup αυτού του τμήματος, τότε το συνολικό speedup είναι ίσο με $\frac{1}{(1-p)+\frac{p}{s}}$ και δεν μπορεί να είναι μεγαλύτερο από $\frac{1}{1-p}$.

## Sensitivity analysis

Έπειτα από την προεπεξεργασία δεδομένων και τη βελτιστοποίηση του πηγαίου κώδικα, προκύπτει ένα νέο και βελτιωμένο μοντέλο του Ογκοπροσομοιωτή Νεφροβλαστώματος, τόσο όσον αφορά στην είσοδο και στο μέγεθός της όσο και αναφορικά με τον τρόπο με τον οποίο υλοποιείται ο αλγόριθμος, οι υπολογισμοί του οποίου εκτελούνται από πολλαπλά threads. Σε αυτό το σημείο, είναι σημαντικό να βεβαιωθεί ότι η λειτουργία της καινούργιας αυτής εκδοχής του μοντέλου είναι η αναμενώμενη και ότι η συνολική συμπεριφορά του μοντέλου παραμένει η ίδια με αυτή του αρχικού verified μοντέλου, πριν την εφαρμογή των αλλαγών στα δεδομένα εισόδου και στον πηγαίο κώδικα. Για αυτό το σκοπό, έπειτα από το στάδιο της προεπεξεργασίας δεδομένων και της βελτιστοποίησης του πηγαίου κώδικα, εφαρμόστηκε ένα βήμα για ανάλυση ευαισθησίας του νέου βελτιωμένου μοντέλου που προκύπτει, έτσι ώστε να βεβαιωθεί ότι η συμπεριφορά του είναι η αναμενώμενη και όμοια με αυτή του αρχικού verified μοντέλου.

Η ανάλυση ευαισθησίας είναι η διαδικασία εξερεύνησης των παραμέτρων εισόδου του μοντέλου, κατά την οποία εκτελούνται προσομοιώσεις με διαφορετικές τιμές για την από κοινού κατανομή τους και αξιολογούνται τα διαφορετικά αποτελέσματα της προσομοίωσης. Έτσι, προκύπτει μία διαίσθηση σχετικά με τον τρόπο που επηρεάζεται η έξοδος του μοντέλου και το αποτέλεσμα της προσομοίωσης από τις διαφορετικές εισόδους και τιμές για τις παραμέτρους του μοντέλου. Το αποτέλεσμα της προσομοίωσης καταγράφεται από τις παραμέτρους εξόδου του μοντέλου, οι οποίες εκφράζουν το ποσοστό μείωσης του όγκου, καθώς και το ιστολογικό του προφίλ, όπως αυτό περιγράφεται από το ποσοστό με το οποίο κάθε κατηγορία κυττάρων, proliferative, dormant, differentiated και dead, συμμετέχει στην αρχική και την τελική σύστασή του. Οι παράμετροι εξόδου της προσομοίωσης παίρνουν εν γένει διαφορετικές τιμές ανάλογα με τις τιμές των παραμέτρων εισόδου του μοντέλου, εκφράζοντας έτσι διαφορετικό αποτέλεσμα της προσομοίωσης για διαφορετικές εισόδους. Τελικά, η αξιολόγηση των αποτελεσμάτων των προσομοιώσεων της ανάλυσης ευαισθησίας προσφέρει μία διαίσθηση σχετικά με τον τρόπο που η είσοδος του μοντέλου επηρεάζει την έξοδό του, συμβάλλει στην κατανόηση της λειτουργίας του και αποτελεί ένα εργαλείο για τον έλεγχο της συνολικής του συμπεριφοράς.

Σε αυτή την εργασία, η ανάλυση ευαισθησίας χρησιμοποιήθηκε για τον έλεγχο της συμπεριφοράς της νέας βελτιωμένης όσον αφορά στην απόδοση της εκτέλεσης εκδοχής του Ογκοπροσομοιωτή Νεφροβλαστώματος, όπως αυτή προκύπτει έπειτα από την προεπεξεργασία της εισόδου και βελτιστοποιήσεις στον πηγαίο κώδικα. Για αυτό το σκοπό, κρίθηκε πιο κατάλληλη η χρήση μίας γραφικής μεθόδου για ανάλυση ευαισθησίας, η οποία απεικονίζει την εξάρτηση των παραμέτρων εξόδου του μοντέλου από τις παραμέτρους εισόδου του μοντέλου σε μία σειρά από scatter plots, προσφέροντας έτσι μία άμεση οπτικοποίηση της επίδρασης της εισόδου της προσομοίωσης στην έξοδό της, με βάση την οποία μπορεί να γίνει εύκολα η αξιολόγηση της συμπεριφοράς του μοντέλου. Πιο συγκεκριμένα, εκτελέστηκε μία σειρά προσομοιώσεων για κάθε παράμετρο εισόδου, διατηρώντας τις τιμές των υπόλοιπων παραμέτρων εισόδου σταθερές στην τιμή αναφοράς και αλλάζοντας την τιμή μόνο της εξερευνούμενης παραμέτρου σε ένα παράθυρο γύρω από την τιμή αναφοράς της. Οι παράμετροι που επιλέχθηκαν για την οπτικοποίηση των αποτελεσμάτων της ανάλυσης σε scatter plots είναι το ποσοστό διαφοροποίησης του μεγέθους του όγκου, το ποσοστό με το οποίο κάθε κατηγορία κυττάρων, proliferative, dormant, differentiated και dead, συμμετέχει στην αρχική σύσταση του όγκου, καθώς και το ποσοστό διαφοροποίησης της συμμετοχής κάθε κατηγορίας στην τελική σύσταση του όγκου σε σχέση με την αρχική.

Αν από αυτή τη γραφική μέθοδο για ανάλυση ευαισθησίας προκύπτει ότι η βελτιωμένη εκδοχή του Ογκοπροσομοιωτή Νεφροβλαστώματος έχει την αναμενώμενη συμπεριφορά, η οποία είναι ίδια συγκριτικά με τη συμπεριφορά του αρχικού verified μοντέλου, πριν την προεπεξεργασία των δεδομένων εισόδου και τις αλλαγές στον πηγαίο κώδικα, τότε η σωστή λειτουργία του νέου βελτιωμένου μοντέλου επιβεβαιώνεται και είναι εφικτή η ασφαλής του χρήση. Σε αντίθετη περίπτωση, πρέπει να γίνει έλεγχος για λάθη στα προηγούμενα βήματα του workflow, π.χ. στον τρόπο με τον οποίο γίνεται η παραλληλοποίηση των υπολογισμών του αλγορίθμου όπου το επιτρέπουν οι εξαρτήσεις δεδομένων, καθώς και ενέργειες για τη διόρθωση των λαθών που εντοπίζονται και επανάληψη της ανάλυσης ευαισθησίας μέχρι η συμπεριφορά του μοντέλου που προκύπτει να είναι η αναμενώμενη. Ενδεικτικά, αναφέρεται ότι οι παράμετροι εισόδοι που επηρεάζουν περισσότερο την έξοδο της προσομοίωσης είναι οι παράμετροι $CKR$, $P_{sym}$ και $P_{sleep}$, οι οποίες εκφράζουν το ποσοστό των κυττάρων που χτυπάει η θεραπεία, το ποσοστό των κυττάρων που εκτελούν συμμετρική διαίρεση και πολλαπλασιάζονται επ' άπειρον και το ποσοστό των κυττάρων που εισέρχονται στην κοιμώμενη $G_0$ φάση εκτός του κυτταρικού κύκλου, όπου τα κύτταρα δεν πολλαπλασιάζονται, αντίστοιχα.

## Clinical adaptation

Εφόσον έχει επιβεβαιωθεί η σωστή λειτουργία του βελτιωμένου μοντέλου του Ογκοπροσομοιωτή Νεφροβλαστώματος, όπως αυτό προκύπτει έπειτα από την προεπεξεργασία των δεδομένων εισόδου και τη βελτιστοποίηση του πηγαίου κώδικα, το νέο μοντέλο μπορεί να αξιοποιηθεί με ασφάλεια για τη χρήση πολλαπλών εικονικών ασθενών για κάθε φυσικό ασθενή, έτσι ώστε η in silico αναπαράσταση του ασθενή να είναι πιο ακριβής και οι προβλεψεις του μοντέλου σχετικά με την εξέλιξη της ασθένειας πιο αξιόπιστες. Αυτή η στρατηγική των πολλαπλών εικονικών ασθενών ανά φυσικό ασθενή αξιοποιήθηκε στο τελικό βήμα του workflow που υλοποιήθηκε σε αυτή την εργασία, κατά το οποίο δημιουργήθηκε ένα σύνολο εικονικών ασθενών, δηλαδή ένα σύνολο από διαφορετικές τιμές για την από κοινού κατανομή των παραμέτρων εισόδου του μοντέλου, για κάθε έναν από τους τρεις φυσικούς ασθενείς του συνόλου δεδομένων, με σκοπό την εκτέλεση μίας σειράς προσομοιώσεων για την κλινική προσαρμογή της CKR παραμέτρου για κάθε εικονικό ασθενή, έτσι ώστε το αποτέλεσμα της προσομοίωσης για κάθε εικονικό ασθενή να εκπροσωπεί βέλτιστα τα πραγματικά κλινικά αποτελέσματα του αντίστοιχου φυσικού ασθενή. Έτσι, προκύπτει τελικά μία κατανομή πιθανότητας για την παράμετρο CKR, ως ζεύγος μέσης τιμής και τυπικής απόκλισης, για κάθε διακριτό ιστολογικό προφίλ και αντίστοιχη ομάδα κινδύνου που εκπροσωπούν οι τρεις ασθενείς του συνόλου δεδομένων.

Στο πρώτο βήμα του αλγορίθμου της κλινικής προσαρμογής δημιουργούνται $N$ εικονικοί ασθενείς με βάση τα απεικονιστικά δεδομένα του φυσικού ασθενή για τον οποίο γίνεται η κλινική προσαρμογή. Ακόμα, κάθε εικονικός ασθενής παίρνει μία διακριτή τιμή για την από κοινού κατανομή των παραμέτρων εισόδου του μοντέλου που δεν καθορίζονται μονοσήμαντα με βάση τα ιατρικά δεδομένα του φυσικού ασθενή. Πιο συγκεκριμένα, κάθε παράμετρος εισόδου παίρνει ξεχωριστά από τις υπόλοιπες παραμέτρους μία τυχαία τιμή με βάση μία προκαθορισμένη κατανομή πιθανότητας. Σε αυτή την εργασία χρησιμοποιήθηκε η ομοιόμορφη κατανομή σε ένα παράθυρο γύρω από την τιμή αναφοράς της εκάστοτε παραμέτρου. Έτσι, δημιουργούνται $N$ in silico εκδοχές του φυσικού ασθενή, κάθε μία από τις οποίες εκπροσωπεί μία εν γένει διαφορετική εκδοχή του προσομοιούμενου βιολογικού συστήματος, η οποία λειτουργεί και εκτελεί τις προσομοιούμενες βιολογικές διεργασίες με βάση διαφορετικές τιμές παραμέτρων. Μετά τη δημιουργία των εικονικών ασθενών, εκτελούνται διαδοχικές προσομοιώσεις για κάθε έναν από αυτούς, με σκοπό την εξερεύνηση της παραμέτρου προσαρμογής, η οποία σε αυτή την εργασία επιλέχθηκε να είναι η $CKR$ παράμετρος, η οποία καθορίζει το ποσοστό τον κυττάρων που χτυπάει η θεραπεία, εκφράζοντας έτσι την αποτελεσματικότητά της. Πιο συγκεκριμένα, για κάθε εικονικό ασθενή εκτελούνται προσομοιώσεις με διαφορετικές τιμές για την $CKR$ παράμετρο, μέχρι τα αποτελέσματα της προσομοίωσης να ταιριάζουν με τα πραγματικά κλινικά αποτελέσματα του αντίστοιχου φυσικού ασθενή, οπότε η τιμή της $CKR$ παραμέτρου για την οποία προκύπτουν αυτά τα αποτελέσματα προσομοίωσης επιλέγεται ως η τιμή βέλτιστης προσαρμογής για τον αντίστοιχο εικονικό ασθενή.

Το κριτήριο με βάση το οποίο αξιολογείται η καταλληλότητας της προσαρμογής αναφέρεται στο ποσοστό μείωσης του μεγέθους του όγκου και στο κατά πόσο αυτό αντιπροσωπεύει την πραγματική κλινική μείωση του όγκου. Θεωρούμε ότι η ανεκτή απόκλιση του ποσοστού μείωσης του όγκου, όπως προκύπτει από την προσομοίωση, σε σχέση με το ποσοστό της πραγματικής κλινικής μείωσης του όγκου, είναι 5%. Παρόλα αυτά, η δυνατότητα ελαχιστοποίησης αυτού του σφάλματος εξαρτάται από την επιλογή των τιμών για τις υπόλοιπες παραμέτρους εισόδου, η οποία διαφέρει μεταξύ των εικονικών ασθενών. Η προσαρμογή της $CKR$ παραμέτρου γίνεται με αναφορά στο διάστημα εξερεύνησης της παραμέτρου, το οποίο αρχικά είναι το $[0, 1]$, αφού η $CKR$ παράμετρος εκφράζει πιθανότητα. Πιο συγκεκριμένα, εκτελείται μία προσομοίωση με τιμή για τη $CKR$ παράμετρο τη μέση τιμή του διαστήματος εξερεύνησης (αρχικά $CKR = 0.5$). Αν το ποσοστό μείωσης του όγκου είναι μικρότερο από το πραγματικό, τότε πρέπει να γίνει εξερεύνηση μεγαλύτερων τιμών, ενώ αν το ποσοστό μείωσης του όγκου είναι μεγαλύτερο από το πραγματικό, τότε πρέπει να γίνει εξερεύνηση μικρότερων τιμών. Έτσι, το διάστημα εξερεύνησης της παραμέτρου προσαρμόζεται συνεχώς, μέχρι να βρεθεί η τιμή που ικανοποιεί βέλτιστα το κριτήριο προσαρμογής για τον εκάστοτε εικονικό ασθενή.

Έτσι, για κάθε εικονικό ασθενή, ο οποίος αποτελεί μία μοναδική in silico εκδοχή του φυσικού ασθενή με εν γένει διακριτή τιμή για την από κοινού κατανομή των παραμέτρων εισόδου του μοντέλου, επιλέγεται μία τιμή για την $CKR$ παράμετρο, η οποία περιγράφει βέλτιστα τα πραγματικά κλινικά αποτελέσματα του φυσικού ασθενή. Ενδέχεται κάποιοι εικονικοί ασθενείς να έχουν τιμές για τις υπόλοιπες παραμέτρους εισόδου που δεν επιτρέπουν την επαρκή κλινική προσαρμογή με σφάλμα σε σχέση με τα πραγματικά κλινικά αποτελέσματα μικρότερο από 5%. Αυτοί οι εικονικοί ασθενείς εξαιρούνται από τη στατιστική ανάλυση που ακολούθησε. Τελικά, έχουμε ένα δειγματικό χώρο από επαρκώς κλινικά προσαρμοσμένες προσομοιώσεις, με μέγιστο πληθικό αριθμό ίσο με το πλήθος των εικονικών ασθενών $N$. Με βάση αυτό το δειγματικό χώρο προκύπτει η κατανομή πιθανότητας για την παράμετρο προσαρμογής, δηλαδή την $CKR$ παράμετρο, για κάθε έναν από τους τρεις φυσικούς ασθενείς του συνόλου δεδομένων και τις αντίστοιχες ομάδες κινδύνου που εκπροσωπούν, ως ζεύγος μέσης τιμής και τυπικής απόκλισης. Επίσης, η ιστολογία του όγκου δεν περιλαμβάνεται στο κριτήριο προσαρμογής του αλγορίθμου, όμως μπορεί να αξιολογηθεί στο τελικό αποτέλεσμα, για να προσφέρει μία ακόμα πιο προσαρμοσμένη στα κλινικά δεδομένα του εκάστοτε φυσικού ασθενή κατανομή πιθανότητας για τη $CKR$ παράμετρο, μειώνοντας περαιτέρω το δειγματικό χώρο των προσομοιώσεων.

## Αποτελέσματα

### Data preprocessing

Στο στάδιο της προεπεξεργασίας δεδομένων δοκιμάζονται διάφορες τιμές για το scale factor, δηλαδή τον πολλαπλασιαστικό παράγοντα που ρυθμίζει το μέγεθος του γεωμετρικού κυττάρου του τρισδιάστατου πλέγματος που παριστάνει in silico το φυσικό όγκο και τον κανονικό ιστό που τον περιβάλλει. Πιο συγκεκριμένα, δοκιμάστηκαν τιμές του scale factor από 1 μέχρι και 10 για κάθε φυσικό ασθενή του συνόλου δεδομένων.

Με την αύξηση του scale factor, αυξάνεται το μέγεθος του γεωμετρικού κυττάρου και μειώνεται έτσι το μέγεθος της εισόδου και το συνολικό κόστος της προσομοίωσης. Πράγματι, η αύξηση του scale factor μείωσε το συνολικό χρόνο εκτέλεσης κα για του τρεις φυσικούς ασθενείς του συνόλου δεδομένων. Πιο συγκεκριμένα, για άυξηση του scale factor από 1 σε 2 ο συνολικός χρόνος εκτέλεσης της προσομοίωσης μειώνεται σημαντικά, για αύξηση του scale factor από 2 σε 3 η μείωση του συνολικού χρόνου εκτέλεσης είναι μικρότερη, ενώ για αύξηση του scale factor από 3 και πάνω η μείωση του συνολικού χρόνου εκτέλεσης δεν είναι αισθητή.

Ταυτόχρονα, η αύξηση του μεγέθους του γεωμετρικού κυττάρου με την αύξηση του scale factor μειώνει την ανάλυση της προσομοίωσης, γεγονός που εισάγει σφάλμα στα αποτελέσματά της. Το σφάλμα αυτό υπολογίστηκε αναφορικά με το ποσοστό μείωσης του μεγέθους του όγκου και παρατηρήθηκε πως όσο η τιμή του scale factor αυξάνεται το σφάλμα του ποσοστού μείωσης του μεγέθους του όγκου σε σχέση αυτό που προκύπτει από τη προσομοίωση με scale factor 1 και την αρχική μέγιστη ανάλυση είναι διάφορο του μηδενός, αλλά και πάλι πολύ μικρό. Παρόλα αυτά, η μείωση της ανάλυσης της προσομοίωσης με την αύξηση του μεγέθους του γεωμετρικού κυττάρου γίνεται αισθητή και από τη 3D ανακατασκευή και απεικόνιση του όγκου κατά τη διάρκεια της προσομοίωσης για τις διάφορες τιμές του scale factor, από την οποία φάνηκε ότι τιμές του scale factor πάνω από 3 είναι απαγορευτικές.

Οι τυπικές τιμές για το μέγεθος του γεωμετρικού κυττάρου, είναι από $1mm$ μέχρι και $3mm$. Το αρχικό μέγεθος του γεωμετρικού κυττάρου είναι ίσο με $1mm$ για κάθε μία από τις εισόδους που αντιστοιχούν στους τρεις φυσικούς ασθενείς του συνόλου δεδομένων. Για δύο από τους ασθενείς επιλέχθηκε scale factor ίσο με 2, με αποτέλεσμα τη μείωση του μεγέθους εισόδου κατά 78%. Για τον άλλο ασθενή επιλέχθηκε scale factor ίσο με 3, με αποτέλεσμα τη μείωση του μεγέθους εισόδου κατά 93%. Σε κάθε περίπτωση, ο συνολικός χρόνος εκτέλεσης μειώθηκε περισσότερο από 97%.

### Profiling & Code optimization

Έπειτα από τη βελτίωση της απόδοσης της εκτέλεσης μέσω της προεπεξεργασίας των δεδομένων εισόδου και τη ρύθμιση του μεγέθους της, έγινε προσπάθεια για τη βελτίωση της απόδοσης με αλλαγές στον πηγαίο κώδικα, έτσι ώστε να εκτελούνται παράλληλα οι υπολογισμοί του αλγορίθμου, όπου το επιτρέπουν οι εξαρτήσεις δεδομένων. Σε αυτό το στάδιο έγινε και το profiling της εκτέλεσης της προσομοίωσης με το massif εργαλείο, το οποίο έδωσε μία εκτίμηση σχετικά με τις απαιτήσεις σε χώρο στη μνήμη, καθώς και με το callgrind εργαλείο, το οποίο έδωσε μία εκτίμηση σχετικά με την κατανομή του κόστους της προσομοίωσης στα διάφορα τμήματα της εκτέλεσης.

Το profiling με το massif εργαλείο έδειξε ότι το στάδιο της προεπεξεργασίας των δεδομένων εισόδου του μοντέλου μειώνει τις απαιτήσεις σε χώρο στη μνήμη κατά 90% περίπου. Ακόμα, η καταγραφή του συνολικού χρόνου εκτέλεσης, καθώς και της ανάλυσής του στα επιμέρους τμήματα/σκαναρίσματα του αλγορίθμου, επιβεβαίωσε ότι η προεπεξεργασία των δεδομένων εισόδου του μοντέλου προκαλεί τη μείωση του χρόνου εκτέλεσης και μάλιστα φάνηκε ότι περισσότερο ευνοείται το κομμάτι της προσομοίωσης στο ιστικό επίπεδο, δηλαδή το τέταρτο, πέμπτο και έκτο σκανάρισμα της προσομοίωσης. Αυτό δικαιολογείται από το γεγονός ότι η προσομοίωση στο ιστικό επίπεδο εκτελεί τη ροή βιολογικών κυττάρων μεταξύ γειτονικών γεωμετρικών κυττάρων για την αύξηση ή τη σμίκρυνση του όγκου, διαδικασία που απαιτεί πολλαπλές προσπελάσεις του τρισδιάστατου πλέγματος του όγκου. Έτσι, είναι λογικό η μείωση του μεγέθους της εισόδου που προκαλείται από την προεπεξεργασία των δεδομένων εισόδου να ευνοεί περισσότερο αυτό το κομμάτι της προσομοίωσης.

Οι εξαρτήσεις δεδομένων επιτρέπουν την εκτέλεση παράλληλων υπολογισμών μόνο για το τρίτο σκανάρισμα που ολοκληρώνει την προσομοίωση στο κυτταρικό επίπεδο για κάθε χρονικό βήμα του αλγορίθμου. Έτσι, μόνο το τρίτο σκανάρισμα ευνοείται από τους βελτιωμένους πόρους, δηλαδή τα πολλαπλά threads που εκτελούν υπολογισμούς ανεξάρτητα και παράλληλα. Αυτό σημαίνει ότι το στάδιο της προεπεξεργασίας των δεδομένων εισόδου, πέρα από το speedup που εισάγει άμεσα λόγω της μείωσης του μεγέθους της εισόδου, εξισορροπεί το φόρτο εργασίας και το κάνει λιγότερο βαρύ στο κομμάτι της προσομοίωσης που αφορά στο ιστικό επίπεδο, βελτιώνοντας έτσι τις προοπτικές για speedup από τη βελτιστοποίηση του πηγαίου κώδικα για αξιοποίηση πολλαπλών threads, σύμφωνα με το νόμο του Amdahl.

Το profiling με το callgrind εργαλείο έδειξε ότι το τρίτο σκανάρισμα συμμετέχει στο συνολικό κόστος της εκτέλεσης της προσομοίωσης σε ποσοστό $0.6\% - 4\%$ πριν την προεπεξεργασία των δεδομένων και σε ποσοστό $13\% - 20\%$ μετά την προεπεξεργασία των δεδομένων. Έτσι, οι αλλαγές που έγιναν στον πηγαίο κώδικα με τη βοήθεια του OpenMP API, για την παράλληλη εκτέλεση των υπολογισμών του τρίτου σκαναρίσματος από πολλαπλά threads, εισάγουν speedup περίπου $16\% - 24\%$, ενώ χωρίς το στάδιο της προεπεξεργασίας των δεδομένων εισόδου το speedup από τη βελτιστοποίηση του πηγαίου κώδικα θα περιοριζόταν στο 4%. Σε κάθε περίπτωση, το πιο σημαντικό speedup εισάγεται άμεσα από το στάδιο της προεπεξεργασίας των δεδομένων εισόδου.

## Sensitivity analysis

Έπειτα από το στάδιο της προεπεξεργασίας των δεδομένων εισόδου της προσομοίωσης και το στάδιο της βελτιστοποίησης του πηγαίου κώδικα, προκύπτει ένα νέο βελτιωμένο μοντέλο, τόσο όσον αφορά στο μέγεθος της εισόδου όσο και αναφορικά με τον τρόπο με τον οποίο εκτελούνται οι υπολογισμοί παράλληλα. Έτσι, προκύπτει ένα βελτιωμένο όσον αφορά στην απόδοση της εκτέλεσης μοντέλο, του οποίου όμως η σωστή λειτουργία πρέπει να ελεγχθεί και να επιβεβαιωθεί. Για αυτό το σκοπό, σε αυτό το σημείο του workflow έγινε η ανάλυση ευαισθησίας του μοντέλου, όπως προκύπτει έπειτα από τις παρεμβάσεις των προηγούμενων βημάτων.

Πιο συγκεκριμένα, εκτελέστηκε η εξερεύνηση κάθε παραμέτρου εισόδου της προσομοίωσης και του τρόπου με τον οποίο επηρεάζει την έξοδο του μοντέλου ξεχωριστά. Για την αξιολόγηση των αποτελεσμάτων χρησιμοποιήθηκε μία γραφική μέθοδος, που οπτικοποιεί τα αποτελέσματα σε scatter plots, αναφορικά με το ποσοστό διαφοροποίησης του μεγέθους του όγκου, την αρχική ιστολογία του όγκου, όπως εκφράζεται από το ποσοστό με το οποίο κάθε κατηγορία κυττάρων, proliferative, dormant, differentiated και dead, συμμετέχει στη σύστασή του, καθώς και τη διαφοροποίηση της ιστολογίας του όγκου, όπως εκράζεται από το ποσοστό διαφοροποίησης του πλήθους των κυττάρων των διαφορετικών κατηγοριών που τον αποτελούν.

Όσον αφορά στη διαφοροποίηση του μεγέθους του όγκου, φάνηκε ότι αυτή επηρεάζεται κατά βάση από την τιμή της $CKR$ παραμέτρου, η οποία καθορίζει το ποσοστό των κυττάρων που χτυπάει η θεραπεία. Πιο συγκεκριμένα, η αύξηση της τιμής της $CKR$ παραμέτρου προκαλεί μεγαλύτερη μείωση του μεγέθους του όγκου. Αυτό είναι λογικό, εφόσον όσο μεγαλύτερη είναι η τιμή της $CKR$ παραμέτρου τόσο περισσότερα βιολογικά κύτταρα εξολοθρεύονται από τη θεραπεία και τόσο πιο αποτελεσματική είναι η θεραπεία συνολικά.

Όσον αφορά στην αρχική σύσταση του όγκου, φάνηκε ότι αυτή επηρεάζεται κατά βάση από τις τιμές των παραμέτρων $P_{sym}$ και $P_{sleep}$, οι οποίες καθορίζουν το ποσοστό των κυττάρων που εκτελούν συμμετρική διαίρεση και το ποσοστό των κυττάρων που εισέρχονται στην κοιμώμενη $G_0$ φάση εκτός του κυτταρικού κύκλου. Πιο συγκεκριμένα, αύξηση της τιμής της $P_{sym}$ παραμέτρου αντιστοιχεί σε επιθετική ανάπτυξη του όγκου με ιστολογία που αποτελείται από περισσότερα proliferative και λιγότερα differentiated κύτταρα. Αντίθετα, αύξηση της τιμής της $P_{sleep}$ παραμέτρου αντιστοιχεί σε πιο ευνοϊκή ιστολογία με λιγότερα proliferative και περισσότερα differentiated κύτταρα. Αυτά τα αποτελέσματα είναι λογικά, καθώς τα κύτταρα που εκτελούν συμμετρική διαίρεση πολλαπλασιάζονται επ' άπειρον, ενώ τα κύτταρα που εισέρχονται στην κοιμώμενη φάση $G_0$ δεν πολλαπλασιάζονται.

Όσον αφορά στην τελική σύσταση του όγκου, φάνηκε ότι αυτή επηρεάζεται από τις τιμές των $CKR$, $P_{sym}$ και $P_{sleep}$ παραμέτρων, με τρόπο που συνάδει με τον τρόπο με τον οποίο αυτές οι παράμετροι επιδρούν στη διαφοροποίηση του μεγέθους του όγκου και στην αρχική του σύσταση. Πιο συγκεκριμένα, αύξηση στις τιμές των παραμέτρων $CKR$ και $P_{sleep}$ αντιστοιχούν σε πιο ευνοϊκή ιστολογία και ανάπτυξη του όγκου, ενώ αύξηση της τιμής της $P_{sym}$ παραμέτρου αντιστοιχεί σε πιο επιθετική ιστολογία και ανάπτυξη του όγκου. Ακόμα, η ιστολογία του όγκου, τόσο αρχική όσο και τελική, επηρεάζεται και από άλλες παραμέτρους, όπως η διάρκεια παραμονής στη φάση $G_0$ ($T_{G0}$), στη νεκρωτική φάση ($T_N$) και στην αποπτωτική φάση ($T_A$), αν και όχι σε τόσο μεγάλο βαθμό.

Όλες αυτές οι παρατηρήσεις αναφορικά με τη συμπεριφορά του νέου βελτιωμένου μοντέλου συνάδουν με τη διαίσθηση που υπάρχει σχετικά με την εξέλιξη των προσομοιούμενων βιολογικών διεργασιών. Ακόμα, συγκρίνοντας της συμπεριφορά του νέου μοντέλου με αυτή που έχει καταγραφεί για το αρχικό verified μοντέλο, βλέπουμε ότι αυτές είναι ουσιαστικά όμοιες. Έτσι, επιβεβαιώνεται η σωστή λειτουργία του νέου μοντέλου, όπως αυτό προκύπτει από τις βελτιώσεις των προηγούμενων βημάτων του workflow, και μπορεί να γίνει η αξιοποίησή του στο τελικό βήμα, για την κλινική προσαρμογή των παραμέτρων.

### Clinical adaptation

Στο τελικό βήμα που υλοποιήθηκε σε αυτή την εργασία, αξιοποιήθηκε η βελτιωμένη όσον αφορά στην απόδοση της εκτέλεσης και verified μέσω ανάλυσης ευαισθησίας εκδοχή του Ογκοπροσομοιωτή Νεφροβλαστώματος για την εκτέλεση πολλαπλών προσομοιώσεων εικονικών ασθενών ανά φυσικό ασθενή και την κλινική προσαρμογή του μοντέλου με βάση τα πραγματικά κλινικά δεδομένα του εκάστοτε φυσικού ασθενή.

Πιο συγκεκριμένα, ο αλγόριθμος της κλινικής προσαρμογής εκτελέστηκε για κάθε φυσικό ασθενή με πλήθος εικονικών ασθενών $N = 20$ και $N = 200$, για την προσαρμογή της $CKR$ παραμέτρου, η οποία εκφράζει την αποτελεσματικότητα της θεραπείας. Η ανάθεση των τιμών για τις υπόλοιπες παραμέτρους εισόδου γίνεται με βάση την ομοιόμορφη κατανομή σε ένα παράθυρο γύρω από την αντίστοιχη τιμή αναφοράς. Η κλινική προσαρμογή θεωρείται επαρκής όταν το ποσοστό μείωσης του μεγέθους του όγκου, όπως προκύπτει από την προσομοίωση, δε διαφφέρει περισσότερο από 5% από το πραγματικό ποσοστό μείωσης του μεγέθους του όγκου, όπως προκύπτει από τα κλινικά δεδομένα του φυσικού ασθενή.

Η κλινική προσαρμογή γίνεται ανεξάρτητα για κάθε εικονικό ασθενή και, έτσι, εκτελέστηκε παράλληλα για το σύνολο των εικονικών ασθενών. Για την κλινική προσαρμογή της $CKR$ παραμέτρου ανά εικονικό ασθενή υλοποιήθηκε ένας αλγόριθμος εξερεύνησης του διαστήματος τιμών της παραμέτρου, ο οποίος εκτελεί μία σειρά από διαδοχικές προσομοιώσεις. Έτσι, ο συνολικός χρόνος εκτέλεσης, για τις παράλληλες εκτελέσεις κλινικών προσαρμογών στο σύνολο των εικονικών ασθενών, εξαρτάται από το χρόνο εκτέλεσης μίας προσομοίωσης, καθώς και από το μέγιστο πλήθος σειριακών εκτελέσεων μεταξύ των εικονικών ασθενών.

Παρατηρήθηκε ότι η αύξηση του πλήθους τον εικονικών ασθενών από 20 σε 200 αύξησε και τη μέση τιμή του πλήθους αυτών των σειριακών εκτελέσεων. Αυτό δικαιολογείται από το γεγονός ότι όσο μεγαλύτερο είναι το πλήθος των εικονικών ασθενών τόσο πιο πολλές εκδοχές του μοντέλου και του in silico βιολογικού συστήματος εκπροσωπούνται όσον αφορά στην τιμή για την από κοινού κατανομή των παραμέτρων εισόδου και, έτσι, είναι πιο πιθανό κάποιες από αυτές τις εκδοχές, δηλαδή κάποιοι από τους εικονικούς ασθενείς, να απαιτούν πιο εκτενή αναζήτηση του διαστήματος εξερεύνησης για την κλινική προσαρμογή της $CKR$ παραμέτρου.

Ακόμα, παρατηρήθηκε ότι η αύξηση του πλήθους των εικονικών ασθενών από 20 σε 200 προκάλεσε αύξηση στο συνολικό χρόνο εκτέλεσης της κλινικής προσαρμογής. Αυτό είναι το φυσικό επόμενο της αύξησης του πλήθους των προσομοιώσεων που εκτελούνται σειριακά, οφείλεται όμως και στο γεγονός ότι οι διαθέσιμοι πυρήνες της αρχιτεκτονικής είναι 48 στο σύνολο και δεν αρκούν για την παράλληλη εκτέλεση της κλινικής προσαρμογής στο σύνολο των 200 εικονικών ασθενών.

Με την ολοκλήρωση της κλινικής προσαρμογής, έχει επιλεχθεί για κάθε εικονικό ασθενή μία τιμή για την $CKR$ παράμετρο, η οποία αποτελεί τη βέλτιστη προσαρμογή που παριστάνει πιο κατάλληλα τα πραγματικά κλινικά αποτελέσματα του αντίστοιχου φυσικού ασθενή για το συγκεκριμένο εικονικό ασθενή. Έτσι, δημιουργούνται δύο δειγματικοί χώροι προσομοιώσεων, ένας που αναφέρεται στο σύνολο των εκτελέσεων και ένας που αναφέρεται μόνο στις προσομοιώσεις βέλτιστης κλινικής προσαρμογής ανά εικονικό ασθενή.

Με βάση αυτούς του δειγματικούς χώρους, έγινε η οπτικοποίηση των κατανομών των παραμέτρων του μοντέλου σε μορφή κανονικοποιημένων ιστογραμμάτων και δόθηκε μία πρώτη εκτίμηση σχετικά με τις τιμές που παριστάνουν επαρκώς τα πραγματικά κλινικά δεδομένα του φυσικού ασθενή. Παρατηρήθηκε ότι η αύξηση του πλήθους των εικονικών ασθενών έδωσε πιο χαρακτηριστική κατανομή για τις παραμέτρους του μοντέλου, εμπλουτίζοντας και βελτιώνοντας έτσι την in silico απεικόνιση του φυσικού ασθενή. Αυτό είναι λογικό, εφόσον η αύξηση του πλήθους των εικονικών ασθενών έχει ως άμεση συνέπεια την αύξηση του πληθικού αριθμού του δειγματικού χώρου των προσομοιώσεων, τόσο συνολικά όσο και για τη βέλτιστη προσαρμογή ανά εικονικό ασθενή.

Η τελική αξιολόγηση των αποτελεσμάτων της κλινικής προσαρμογής έγινε με τη στατιστική ανάλυση των τιμών της $CKR$ παραμέτρου στο δειγματικό χώρο των προσομοιώσεων βέλτιστης κλινικής προσαρμογής, αξιολογώντας ταυτόχρονα την τήρηση του κριτηρίου προσαρμογής. Πιο συγκεκριμένα, η τιμή για την από κοινού κατανομή των παραμέτρων εισόδου που εκπροσωπεί κάποιος εικονικός ασθενής ενδέχεται να μην του επιτρέπει την επαρκή κλινική προσαρμογή στα κλινικά αποτελέσματα του φυσικού ασθενή που παριστάνει in silico, όσο και αν εξερευνηθεί και προσαρμοστεί η $CKR$ παράμετρος. Οι επιλεγμένες για αυτούς του εικονικούς ασθενείς τιμές της $CKR$ παραμέτρου εξαιρέθηκαν από τη στατιστική ανάλυση για την εξαγωγή του τελικού αποτελέσματος.

Το τελικό αποτέλεσμα της κλινικής προσαρμογής είναι η κατανομή πιθανότητας της $CKR$ παραμέτρου, όπως αυτή εκφράζεται από το ζεύγος μέσης τιμής και τυπικής απόκλισης, για κάθε μία από τις τρεις ιστολογικές ομάδες και αντίστοιχες ομάδες κινδύνου που εκπροσωπούνται στο σύνολο δεδομένων. Ακόμα, παρόλο που η ιστολογία του όγκου δε συμπεριλαμβάνεται άμεσα στο κριτήριο προσαρμογής του αλγορίθμου, αυτή αξιολογήθηκε στην εξαγωγή του τελικού αποτελέσματος, με βάση το ποσοστό με το οποίο οι διαφορετικές κατηγορίες κυττάρων, proliferative, dormant, differentiated και dead, συμμετέχουν στην αρχική και τελική σύσταση του όγκου. Η μέση τιμή της $CKR$ παραμέτρου για την ομάδα ευνοϊκής ιστολογίας και χαμηλού κινδύνου υπολογίστηκε ίση με $0.9$ περίπου, για την ομάδα μικτής ιστολογίας και μεσαίου κινδύνου υπολογίστηκε ίση με $0.8$ περίπου, ενώ για την ομάδα δυσμενούς ιστολογίας και υψηλού κινδύνου υπολογίστηκε ίση με $0.5$ περίπου.

## Συμπεράσματα & Επόμενα Βήματα

### Αντιμετώπιση του bottleneck που εισάγει στην απόδοση της εκτέλεσης η προσομοίωση στο ιστικό επίπεδο και αλλαγές στον πηγαίο κώδικα

Το κομμάτι της εκτέλεσης που επιταχύνθηκε σε αυτή την εργασία, με την ταυτόχρονη παράλληλη εκτέλεση των υπολογισμών του από πολλαπλά threads, αφορά στο τρίτο σκανάρισμα του αλγορίθμου και στην προσομοίωση στο κυτταρικό επίπεδο βιοπολυπλοκότητας. Παρόλα αυτά, το κομμάτι της εκτέλεσης που συμμετέχει περισσότερο στο συνολικό κόστος αφορά στο τέταρτο, πέμπτο και έκτο σκανάρισμα του αλγορίθμου και στην προσομοίωση στο ιστικό επίπεδο βιοπολυπλοκότητας. Πιο συγκεκριμένα, ακόμα και μετά την προεπεξεργασία των δεδομένων που μειώνει το μέγεθος της εισόδου, ευνοεί περισσότερο την προσομοίωση στο ιστικό επίπεδο και εξισορροπεί το φόρτο εργασίας, η προσομοίωση στο ιστικό επίπεδο συμμετέχει περισσότερο από 80% στο συνολικό χρόνο εκτέλεσης.

Έτσι, η προσομοίωση στο ιστικό επίπεδο βιοπολυπλοκότητας εισάγει ένα bottleneck όσον αφορά στην απόδοση της εκτέλεσης της προσομοίωσης και στη βελτίωσή της. Η πηγή αυτού του προβλήματος είναι οι εξαρτήσεις δεδομένων που χαρακτηρίζουν τους υπολογισμούς που εκτελούνται στο τέταρτο, πέμπτο και έκτο σκανάρισμα του αλγορίθμου. Πιο συγκεκριμένα, σε αυτά τα σκαναρίσματα υλοποιείται η ροή βιολογικών κυττάρων μεταξύ γειτονικών γεωμετρικών κυττάρων του τρισδιάστατου πλέγματος που παριστάνει in silico το φυσικό όγκο και τον κανονικό ιστό που τον περιβάλλει, για την προσομοίωση της αύξησης ή της σμίκρυνσης του όγκου. Αυτή η ροή βιολογικών κυττάρων προκαλεί πολλές εξαρτήσεις δεδομένων και race conditions και για να αντιμετωπιστεί πρέπει να γίνουν βασικές αλλαγές στη λογική και στην υλοποίηση του αλγορίθμου.

Άλλες αλλαγές που θα μπορούσαν προαιρετικά να γίνουν στον αλγόριθμο του μοντέλου αφορούν στο συνυπολογισμό της ιστολογίας του όγκου κατά τη διάρκεια της προσομοίωσης. Όπως είναι τώρα το μοντέλο, η ιστολογία του όγκου συμπεριλαμβάνεται έμμεσα στην προσομοίωση, με την αναφορά στο ποσοστό με το οποίο κάθε κατηγορία κυττάρων, proliferative, dormant, differentiated και dead, συμμετέχει στην αρχική και τελική σύσταση του όγκου. Παρόλα αυτά, δεν αξιοποιούνται τα ιστοπαθολογικά δεδομένα του συγκεκριμένου φυσικού ασθενή, καθώς τα voxels του τρισδιάστατου πλέγματος του όγκου αρχικοποιούνται είτε ως κανονικός ιστός είτε ως ιστός που ανήκει στον όγκο.

Θα ήταν ενδιαφέρον να δοκιμαστούν επεκτάσεις οι οποίες κωδικοποιούν στο τρισδιάστατο πλέγμα πληροφορία σχετικά με την ιστολογία του όγκου, για παράδειγμα να υπάρχουν βλαστηματικές, στρωματικές και επιθηλιακές περιοχές και σε κάθε μία από αυτές να εφαρμόζονται διαφορετικές τιμές για τις παραμέτρους του μοντέλου. Ακόμα, αντί να ρυθμίζεται στατικά το μέγεθος του γεωμετρικού κυττάρου στο στάδιο της προεπεξεργασίας δεδομένων, θα μπορούσε αυτό να γίνεται δυναμικά κατά την εκτέλεση της προσομοίωσης, ανάλογα με τις υπολογιστικές ανάγκες και για κάθε ιστολογική περιοχή του τρισδιάστατου πλέγματος χωριστά.

### Επιτάχυνση του μοντέλου και δημιουργία μίας αρχιτεκτονικής εκτέλεσης πολλαπλών προσομοιώσεων για την κλινική προσαρμογή του συνόλου των παραμέτρων του μοντέλου

Έπειτα από τη διαμόρφωση του αλγορίθμου και την υλοποίησή του με γνώμονα τον περιορισμό των εξαρτήσεων δεδομένων και τη μεγιστοποίηση των προοπτικών για επιτάχυνση, μπορεί να δημιουργηθεί μία αρχιτεκτονική για την αποδοτική εκτέλεση πολλαπλών προσομοιώσεων εικονικών ασθενών ανά φυσικό ασθενή. Για αυτό το σκοπό, είναι χρήσιμο να εξερευνηθούν οι παράμετροι του αλγορίθμου της κλινικής προσαρμογής, για παράδειγμα του πλήθους των εικονικών ασθενών $N$, έτσι ώστε να προσδιοριστούν οι τιμές τους που δίνουν καλύτερα αποτελέσματα. Έτσι, μπορεί να ξεκινήσει ο προσδιορισμός των προδιαγραφών του υπολογιστικού συστήματος και της αρχιτεκτονικής για εκτέλεση πολλαπλών προσομοιώσεων, έτσι ώστε να πληρούνται κάποια προκαθορισμένα κριτήρια απόδοσης.

Εφόσον έχει δημιουργηθεί η αρχιτεκτονική εκτέλεσης μαζικών προσομοιώσεων, αυτή μπορεί να αξιοποιηθεί για την εκτέλεση πολλαπλών προσομοιώσεων εικονικών ασθενών ανά φυσικό ασθενή, για την κλινική προσαρμογή και των υπόλοιπων παραμέτρων εισόδου του μοντέλου, με αντίστοιχο τρόπο όπως έγινε σε αυτή την εργασία για τη $CKR$ παράμετρο. Σε αυτό το σημείο, τίθεται το ζήτημα που αφορά στην αβεβαιότητα των αντιπροσωπευτικών τιμών αναφοράς για τις παραμέτρους του μοντέλου και στις εξαρτήσεις που υπάρχουν μεταξύ τους. Για να αντιμετωπιστεί αυτό το πρόβλημα, προτείνεται να ξεκινήσει η κλινική προσαρμογή από τις παραμέτρους που επηρεάζουν περισσότερο την έξοδο της προσομοίωσης, δηλαδή τις $CKR$, $P_{sym}$ και $P_{sleep}$ και έπειτα να προσαρμοστούν οι υπόλοιπες παράμετροι, με βάση και τη σχέση τους με το χρόνο διπλασιασμού του όγκου $T_d$.

Αυτή η διαδικασία μπορεί να επαναληφθεί πολλές φορές, μέχρι τα αποτελέσματα της κλινικής προσαρμογής να συγκλίνουν ικανοποιητικά για κάθε παράμετρο. Έτσι, προκύπτουν τελικά κατανομές πιθανότητας για κάθε παράμετρο του μοντέλου, οι οποίες χαρακτηρίζουν τις διακριτές ομάδες ασθενών του συνόλου δεδομένων. Τέλος, είναι σκόπιμο να αυξηθεί το μέγεθος του συνόλου δεδομένων, έτσι ώστε να εκπροσωπούνται σε αυτό περισσότερες κατηγορίες ασθενών και με περισσότερα δείγματα η κάθε μία. Έτσι, οι κατανομές πιθανότητας που προκύπτουν για τις παραμέτρους του μοντέλου μέσω της κλινικής προσαρμογής θα είναι πιο αξιόπιστες και αντιπροσωπευτικές.

### Χρήση της αρχιτεκτονικής πολλαπλών προσομοιώσεων εικονικών ασθενών για τη μεταφορά του Ογκοπροσομοιωτή Νεφροβλαστώματος στην κλινική πρακτική ως συμβουλευτικού εργαλείου που προσφέρει υπηρεσίες εξατομικευμένης ιατρικής

Εφόσον έχουν προσδιοστιστεί οι κατανομές των παραμέτρων εισόδου του μοντέλου για τις διαφορετικές κατηγορίες ασθενών, μπορεί να γίνει η αξιοποίηση της αρχιτεκτονικής εκτέλεσης μαζικών προσομοιώσεων με σκοπό τη χρήση πολλαπλών εικονικών ασθενών για την πρόβλεψη των κλινικών αποτελεσμάτων στην εφαρμογή ποικίλων θεραπευτικών σεναρίων για την αντιμετώπιση του όγκου του Wilms σε τρέχοντες φυσικούς ασθενείς, οι οποίοι δεν έχουν ακόμα δεχθεί θεραπεία. Πιο συγκεκριμένα, ανάλογα με την ιστολογική κατηγορία και αντίστοιχη ομάδα κινδύνου στην οποία ανήκει ο φυσικός ασθενής, μπορεί να εκτελεστεί μία σειρά από προσομοιώσεις εικονικών ασθενών με βάση τα απεικονιστικά του δεδομένα και τιμή για την από κοινού κατανομή των παραμέτρων εισόδου με βάση τις κατανομές που έχουν προκύψει μέσω της κλινικής προσαρμογής για αυτή την κατηγορία ασθενών.

Αξιολογώντας το αποτέλεσμα των μαζικών προσομοιώσεων συνυπολογίζοντας τα αποτελέσματα των επιμέρους εικονικών ασθενών, για παράδειγμα εκτελώντας στατιστική ανάλυση στο σύνολό τους, μπορεί να απεικονιστεί αξιόπιστα ο φυσικός ασθενής στον υπολογιστή και να γίνει μία ακριβής εκτίμηση της κλινικής του πορείας. Εκτελώντας αυτή τη διαδικασία για την προσομοίωση διαφορετικών θεραπευτικών προσεγγίσεων, μπορεί να γίνει η σύγκριση των αποτελεσμάτων για τα διαφορετικά αυτά θεραπευτικά σενάρια και να ξεκινήσει η μεταφορά του Ογκοπροσομοιωτή Νεφροβλαστώματος στην κλινική πρακτική, ως εργαλείου για εξατομικευμένη ιατρική που προσφέρει συμβουλές για την επιλογή του σχεδίου θεραπευτικής αντιμετώπισης της νόσου με τα πιο υποσχόμενα αποτελέσματα. Για παράδειγμα, εκτελώντας την κλινική πρόβλεψη για την ελεύθερη ανάπτυξη του όγκου, καθώς και για τη χορήγηση της προεγχειρητικής χημειοθεραπείας του καθιερωμένου SIOP πρωτοκόλλου, μπορεί να απαντηθεί το απλό ερώτημα που αφορά στον αν πρέπει τελικά να προηγηθεί χημειοθεραπεία πριν από τη χειρουργική αφαίρεση του όγκου.

Η μεταφορά του Ογκοπροσομοιωτή Νεφροβλαστώματος στην κλινική πρακτική πρέπει να γίνει στο πλαίσιο κλινικών δοκιμών που θα βεβαιώσουν τη σωστή και αποτελεσματική του λειτουργία στο ιατρικό περιβάλλον. Τέτοιες κλινικές δοκιμές πρέπει να σχεδιαστούν και να εφαρμοστούν από έμπειρους γιατρούς σε ένα επαρκώς μεγάλο και αντιπροσωπευτικό κλινικό δείγμα, πάντα με σεβασμό και τήρηση των ιατρικών πρωτοκόλλων. Για αυτό το σκοπό, είναι απαραίτητη η διάθεση του μοντέλου στο πλαίσιο μίας ολοκληρωμένης εφαρμογής, η οποία περιλαμβάνει κάποια βασικά δομικά στοιχεία και χαρακτηριστικά και διατίθεται με βάση τις καθιερωμένες πρακτικές, έτσι ώστε να μπορεί να χρησιμοποιείται εύκολα από γιατρούς σε διαφορετικά εν γένει κλινικά κέντρα. Η εφαρμογή αυτή θα πρέπει σίγουρα να περιλαμβάνει ένα DBMS (Database Management System) για την κατάλληλη και αποτελεσματική αποθήκευση και πρόσβαση στα ιατρικά δεδομένα, με τήρηση των κανόνων προστασίας δεδομένων (GDPR requirements), καθώς και μία διεπαφή χρήστη (user interface) που θα επιτρέπει την πραγματοποίηση των επιθυμητών σεναρίων χρήσης της εφαρμογής (use case scenarios), εξασφαλίζοντας μία φιλική προς το χρήστη εμπειρία (user-friendly experience).

# Contents

# List of Figures

# List of Tables

# Introduction

The Oncosimulator is a tool for in silico medicine, and more specifically for in silico oncology. It utilizes the medical data of the physical patient that suffers from a neoplastic disease, in order to represent the solid tumor in silico, i.e. on the computer, as a 3D matrix of voxels and simulate its development in time, either as the tumor grows freely or within the context of a therapeutic scheme. Moreover, the progression of the simulation is governed by a set of simulation input parameters that define the biologic processes at play and the overall attributes of the system that is being simulated, while the outcome of the simulation is expressed by the final size and shape of the tumor, the tumor volume variation and the final composition of the tumor by the different cell type populations, which indicates the tumor's histology.

The simulation of the evolution of the tumor growth is performed in discrete time steps. In that respect, the Oncosimulator falls in the category of discrete entity-discrete event agent-based models for multiscale cancer modeling. Within each time step, a series of consecutive scans of the 3D tumor matrix is performed, for the simulation on the cellular and on the tissue level of biocomplexity. In that respect, the Oncosimulator utilizes the "summarize and jump" technique, which "summarizes" the simulation results on one level of biocomplexity before a "jump" is made to another level of biocomplexity, for each discrete time step of the simulation. Here, the "summarize and jump" process is made between the cellular and tissue levels for each discrete time step of the simulation. Finally, the Oncosimulator is a top-down model, in that it is mainly data-driven rather than based solely on a network of complex components that strictly describe the simulated processes.

The Nephroblastoma Oncosimulator is a version of the Oncosimulator software that is used for the execution of simulations that are specific to Wilms' tumor, also known as nephroblastoma, which is the most common kidney tumor in childhood. It was developed by members of the In Silico Oncology and In Silico Medicine Group, ICCS, ECE, National Technical University of Athens, under the lead of G. Stamatakos and the clinical advisorship of N. Graf, University of Saarland from 2006 to 2019. It simulates the tumor evolution when the standard SIOP protocol management of the disease is applied. The simulated therapeutic scheme consists of a 4-week mixed chemotherapeutic regimen, which includes Vincristine and Actinomycin administration. The parameter that expresses the effects of the administered therapy is the cell kill ration ($CKR$) parameter, which determines the ratio of cells that are hit by therapy. For the Nephroblastoma Oncosimulator, the $CKR$ parameter is analyzed in two parameters, one that corresponds to the effects of the Vincristine drug dose ($CKR\_VCR$) and one that corresponds to the effects of the Actinomycin drug dose ($CKR\_ACT$).

The main goal of the Oncosimulator, for the nephroblastoma paradigm and for every other neoplastic disease specific version, is to represent the tumor along with its normal tissue microenvironment in silico, so that the behavior of the neoplasm can be predicted and recorded for different conditions, e.g. for free growth or for therapy administration within different therapeutic schemes. In that respect, the Oncosimulator acts as a digital twin for the physical tumor and the biologic processes that cause its development in time. That way, it can be utilized for the patient specific study of the tumor and its normal tissue microenvironment and ultimately used as an advisor for personalized medicine, offering guidelines for the best course of the disease management to the medical doctors.

The digital twin terminology refers to the in silico representation of a physical system. It is a popular method in engineering for product design and development and for the monitoring and study of physical systems on the computer. The main characteristic that differentiates the digital twin method from a classical simulation is that it offers an accurate representation of the physical system environment that can be used for the execution of a set of simulations and evaluations of the system's state, via the constant flow of data between the physical system and its digital representation. The Oncosimulator has the features of a digital twin model, since it utilizes the medical data of the patient for the accurate patient-specific representation of the tumor in silico. Even though the patient-specific medical imaging data are easily translated in silico with respect to the size and shape of the tumor, the translation of the characteristics of the tumor histology and of the regulation of the other biologic processes at play is more complex, but crucial for the function of the Oncosimulator as a digital twin for the tumor and its normal tissue microenvironment.

The evolution of the simulation is determined by a set of input parameters that define the biologic processes that occur as the simulation progresses, one of them being the $CKR$ parameter which represents the ratio of cells that are hit by the administered therapy and thus expresses the effects of the therapy. The patient-specific values for these parameters are not unambiguously derived from the medical data of the patient, but are crucial for the accurate patient-specific in silico representation of the tumor. In the past, the values for the simulation input parameters have been determined via sensitivity analysis, i.e. the process of exploring the values for the simulation input parameters, executing the corresponding simulations and evaluating the simulation outcomes and the manner in which they are affected by the variation of the values for the simulation input parameters. However, a more thorough and methodical technique for the determination of the values for the simulation input parameters is required, so that the derived values accurately describe the specific patient, a digital twin for the patient-specific tumor and normal tissue microenvironment is created and, finally, the Oncosimulator acts as a medical advisor for personalized medicine.

The purpose of the present work is to explore high performance computing methods for the Nephroblastoma Oncosimulator, both for the performance enhancement of a single simualtion execution and for the efficient execution of a multitude of simulations for a specific physical patient. The uncertainty that the assignment of values for the simulation input parameters encompasses renders the execution of the Nephroblastoma Oncosimulator multiple times and for different values for the joint distribution of the simulation input parameters essential. That way, a mutlitude of virtual patients is created in silico for each actual physical patient and, executing a simulation for each virtual patient and coevaluating the simulation outcomes, the ambiguousness that accompanies the assignment of values for the simulation input parameters is addressed and a more accurate digital representation of the physical patient and prediction of the actual results of the administered therapy can be provided. If the Nephroblastoma Oncosimulator is to be used as an on-line digital twin model that produces real-time outputs, the satisfaction of carefully designed and strict performance goals along this process, both for a single simulation execution and for the overall execution of a cluster of virtual patient simulations, is crucial, which underlines the role of high performance computing.

Here, the execution of multiple simulations in the context of matching a single physical patient to many in silico virtual patients, each one with a distinct value for the joint distribution of the simulation input parameters, is utilized for the determination of the value for the $CKR$ parameter, which expresses the effectiveness of the administered therapy, for the specific physical patient, a task which has already been performed in the past via sensitivity analysis, but inadequately. More specifically, a set of virtual patients is created for each physical patient from a set of three already treated patients and the $CKR$ parameter is explored for each virtual patient, so that the simulation outcomes match the actual outcomes of the corresponding physical patient. The three patients, whose medical data, including the applied therapeutic scheme, the medical imaging data before and after the therapeutic scheme was applied and the histologic data, as they were determined via the post-operative bioptic check, were provided by the Saarland University Hospital, represent three distinct histologic profiles and corresponding risk groups; low, intermediate and high risk.

The derived values for the $CKR$ parameter for each physical patient indicate the distibution for the value of the $CKR$ parameter for the distinct histologic profiles and corresponding risk groups that are represented in the patient dataset. As the patient dataset grows, this distribution for the value of the $CKR$ parameter becomes richer and more accurate, since more patient groups are present in the dataset and each group is represented by a bigger number of samples. Moreover, the same process can be followed for the determination of the distribution for the values of the rest of the simulation input parameters, with a focus on the parameters that significantly affect the outcome of the simulation and are not referenced adequately in the bibliography, such as the ratio of cells that perform symmetric division $P_{sym}$ and the ratio of cells that enter the quiescent $G_0$ phase $P_{sleep}$. Once the distributions for the values of the simulation input parameters are adequately defined, a complete and integrated application can be designed for the execution of multiple virtual patient simulations for current patients that have not yet been treated, with the values for the simulation input parameters being derived by the corresponding distributions according to the patient group that the current physical patient belongs to. That way, the Nephroblastoma Oncosimulator can be utilized in clinical trials and eventually be used as an advisor for personalized medicine in a clinical setting.

# Theoretical Framework

## 2.1 Multiscale Cancer Modeling

### 2.1.1 Cancer as a multiscale phenomenon and the different scales of biocomplexity that describe it

Cancer is a complex biological phenomenon that is expressed as the manifestation of abnormal cell and tissue behavior, which comes as the result of processes and events that incorporate mutliple levels of biocomplexity.

The basis of cancer is the accumulation of mutations in cells, which results in them becoming cancerous and behaving abnormaly. These mutations are the result of a number of genetic and environmental factors that act on a variety of biological entities and mechanisms inside the organism. There exists a set of rules, known as hallmarks of cancer, that govern the transformation of normal cells into malignant cancers [1]. The major hallmarks of cancer are depicted in **Figure 2.1**. They form the set of traits that all cancerous cells have acquired through a variety of mechanisms, in order to keep proliferating and urging the dynamic growth of the neoplasm.

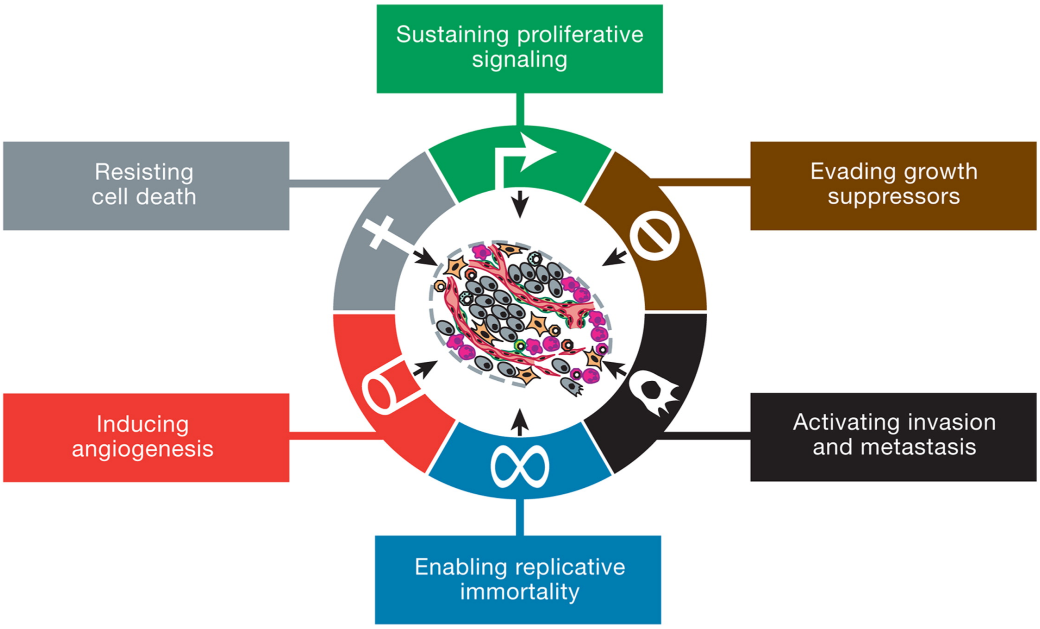

Sustaining proliferative signaling
Resisting cell death
Evading growth suppressors
Inducing angiogenesis
Activating invasion and metastasis
Enabling replicative immortality

**Figure 2.1:** The six major hallmarks of cancer, which summarize the distinct and coplementary capabilities that all cancerous tissue possesses. These capabilities constitute the set of properties that are characteristic to the neoplastic disease and aid its development [1].

The hallmarks of cancer don't concern only the cancerous cells, but all the mechanisms of the organism that take part in the development of the cancer. These mechanisms refer not only to the cancerous neoplasm, but also to its microenvironment of normal and necrotic cells, as well as to biological procedures that happen or are orchestrated in different tissues of the organism. The challenge that the cancer modeling is facing is the adequate modeling of the hallmarks of cancer, combining the biological entities and mechanisms that take part in the cancer growth and in the proper scale of biocomplexity [2].

These different scales of biocomplexity are shown in **Figure 2.2**. The major biocomplexity scales refer to space and time.

The **spatial scales** axis depicts the biocomplexity scaling in space. If the target process that is being modelled refers to interactions between atoms then the scale in space is low. The spatial scale increases sequentially from the atomic level, to the molecular, organelle, cellular, tissue, organ, system, organism and finally to the highest spatial scale, the population. Every scale element has as its structural element the scale element that is one step lower.

The **temporal scales** axis depicts the biocomplexity scaling in time. The chosen time scale depends on the rate of the process that is being modelled. If the process progresses very slowly then the suitable temporal scale is high. In the case of high frequency processes, the chosen temporal scale should be low, in order for the progress to be recorded adequately.

The rest of the axes, **bioprocesses**, **cancer types** and **treatment modalities**, do not refer to the scales of the biocomplexity. They are included for the sake of completeness and to showcase some typical biological processes, cancer types and treatment modalities that are modeled and simulated in silico, using the spatiotemporal scales.

Typically, as the biocomplexity scale in space increases so does the temporal biocomplexity scale. The examples of the atomic scale, the molecular scale, the microscopic scale and the macroscopic scale are presented briefly [3], in order to verify the relationship between spatial and temporal scales and also to familiarize the reader with the notion of biocomplexity scales in cancer modeling.

The **atomic scale** incorporates interactions between atoms and molecules, which are primarily modeled using molecular dynamics (MDs), in order to study the structure and dynamic properties of various macromolecules, such as proteins, peptides and lipids. The spatial scale is in the order of $nm$, while the temporal scale is in the order of $ns$.

The **molecular scale** is primarily used in the simulation of the cell signaling mechanisms by investigating the average of the properties of a population of proteins, rather than the properties of individual proteins, which is investigated in the atomic scale. To this end, ordinary differential equations (ODEs) are used for the modeling (more on modeling techniques in the next paragraph) and the spatial scale is in the order of $nm \sim \mu m$, while the temporal scale is in the order of $\mu s \sim s$.

The **microscopic scale** describes the malignant transformation of normal cells, associated alterations of cell–cell and cell-matrix interactions, the heterogeneous tumor environment and the element of tumor heterogeneity. This is achieved by investigating cell and tissue properties, using partial differential equations (PDEs) and agent based modeling (ABM). The spatial scale is in the order of $\mu m \sim mm$ and the temporal scale is in the order of $min \sim hour$.

Finally, the **macroscopic scale** studies the gross tumor behavior including morphology, shape, extent of vascularization, and invasion, under different environmental conditions. This is achieved by treating some or all of the cells of the microscopic scale as a continuum, in order for cell and substrate transport to be modeled with conservation laws for spatiotemporally-varying densities (i.e., PDEs), rather than keeping track of individual cell activities. The spatial scale is in the order of $mm \sim cm$ and the temporal scale is in the order of $day \sim year$.

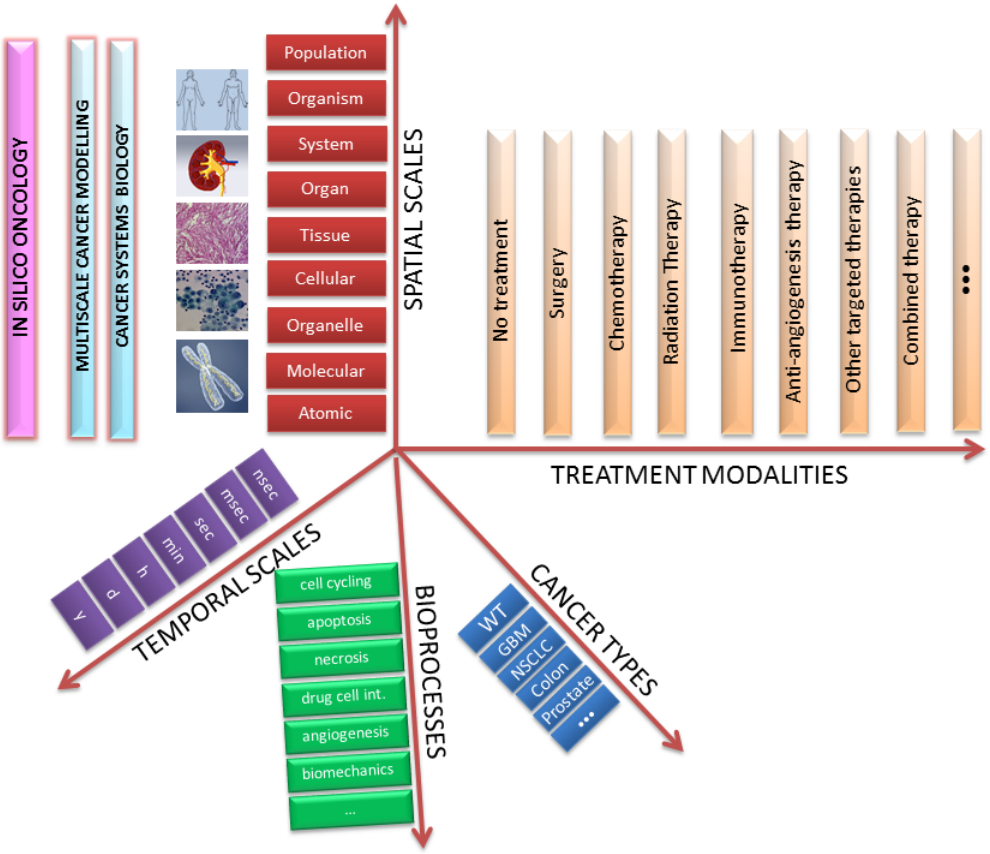


**Figure 2.2:** The scales of biocomplexity in cancer modeling [4]. The callenge that cancer modeling is facing is to represent cancer as a multiscale phenomenon, by modeling the processes that describe it, as defined by the major hallmarks, combining adequately the different spatiotemporal scales of biocomplexity.

### 2.1.2 A comparison of the techniques that are employed for modeling and integrating the various scales of cancer biocomplexity

The existing cancer modeling techniques mainly use mathematical methods that are derived from continuous and discrete mathematics. These mathematical methods aim to put the empirical medical and scientific knowledge on predictably regulated biological processes and events in the exact and strict mathematical context. The prefered major mathematical method, continuous or discrete, depends on the application profile, i.e. the target processes that the application aims to simulate and explore in silico and the spatiotemporal scales that are inherent to these processes.

The **continuous method** utilizes techniques that come from the fields of mathematical analysis and differential equations, in order to express the biological problems in terms of systems of equations and conditions of continuous variables. This method has been used in many applications, such as the description of molecular dynamics in the context of cancer mutations within the epidermal growth factor receptor (EGFR) [5], the description of tumor growth in the form of the universal law in the context of radiotherapeutic schemes [6] and the description of tumor growth using the diffusion differential equation [7, 8, 9], among others. The challege that this method introduces is the adequate trasnlation of the continuous analytical models to their equivalent arithmetic approximates that can be ran in silico, i.e. on the discrete computing system. The verification of the final resulting model, in order to ensure that it implements an arithmetic method that converges and has results that are adequately similar to the ones that the analytical model produces, is crucial and necessary before the use of the model with medical data as its input [10].

The **discrete method** takes advantage of the discrete nature of the various biological entities, which are organized in discrete quanta, as shown in the spatial scales axis in **Figure 2.2**. This method aims to simulate these discrete biological phenomena using discrete techniques that are inspired by algorithmic logic, such as cellular automata [11, 12, 13] and agent-based modeling [14, 15, 16, 17]. As opposed to the continuous method, the discrete method has the advantage that the final model can easily and in a straightforward manner be traslated to computer code and be realized as an in silico application. The Oncosimulator is a typical example of a model that uses the discrete method, implementing the cellular automata and agent-based modeling concepts using the hypermatrix notation, in order to simulate the processes and events that take place at the cellular spatial scale of biocomplexity. The hypermatrix notation and the Oncosimulator as a discrete model are presented in detail in the Oncosimulator dedicated chapter (**Chapter 2.3**), where they can be put in better context.

There also exist **hybrid models** that use both the continuous method and the discrete method, in order to integrate the different biocomplexity scales for the simulation of a range of bioprocesses. That way, the complex phenomenon of cancer can be better simulated and explored, via the simulation of the various hallmarks of cancer (**Figure 2.1**), of the mechanisms that are behind them and of the way that they interact. A typical example of a hybrid model is the Oncosimulator of the Contra Cancrum project, which, in the context of glioblastoma and lung cancer paradigms, uses the hypermatrix notation in order to implement the neoplasm in a discrete 3D mesh, but also uses the continuous diffusion equation with the appropriate boundary conditions for the spatial growth of the tumor [18, 19, 20, 21, 7, 8].

There exist two primary approaches for the integration of the different biocomplexity scales, the bottom-up approach and the top-down approach [22].

The integration of the different biocomplexity scales has typically been performed using the so called **bottom-up approach**. This approach aims to study and explore the target system via the methodical simulation of its core components moving from the lowest to the highest spatial biocomplexity scales that exist in it. The target system can in that way be assembled from its known core parts, e.g. proteins and other macromolecules, which in turn form the spatial scale that is one step higher in the spatial scale axis, e.g. cells, until the target system, e.g tissues or organs, is implemented. The bottom-up approach is a scientific method that models the target biological system in a manner that is exact and methodical, in order to explore, record and ultimately understand the bioprocesses that take place in it. However, this method presupposes that the core components of the target biological system to be modeled, as well as the way that they interact and the biological processed at play are already known and well-defined, which is often not the case in systems biology, especially when studying the complex human organism and more so when studying the complex phenomenon of cancer. As a result, a disadvantage that is inherent to the bottom-up approach is the fact that its time a new component is discovered and added to the model, the whole model needs to be reconfigured.

In order to overcome the limitations that the bottom-up approach introduces, an alternative approach was devised, the **top-down approach**. The top-down approach is in many ways the opposite of the bottom-up approach. It uses medical data, in order to enrich the model with the empirical clinical observation. These data are used along with the general medical knowledge in a dynamic iterative process that aims to find the highest granularity of the system that will make the in silico model as accurate as possible in reflecting reality. As a result, the model is not solely dependent on the existing scientific knowledge, but can also take advantage of the dynamic acquisition of a wealth of biomedical data. These data are inherent to real patients, which makes the top-down approach ideal for applications of personalized medicine. A typical example of a such application is the Oncosimulator, which is presented in detail in **Chapter 2.3**, where it can be put in better context. As opposed to the bottom-up approcah, the top-down approach structures the model so that the highest spatial scale is implemented first with reference to the medical data of the model and the lower scales follow. The transition from one scale to the other is performed using the **summarize and jump strategy** [23, 24]. As its name indicates, this strategy is based on the idea of summarizing the events that take place in one scale of biocomplexity before jumping to the next scale, which makes use of the summary produced by the previous scale.

The top-down approach overcomes the limitations and disadvantages of the bottom-up approach primarily by making use of data instead of multiple components that form a hypercomplex network. The prefered approach depends on the target application. The top-down approach is more fit for personalized medicine applications and for practical clinical applications in general, while the bottom-up approach is more fit for the simulation of well-defined and idealy simplistic bioprocesses, in order to validate the existing scientific knowledge on them or to be used as components of more complex models.

## 2.2 The Eukaryotic Cell Cycle

### 2.2.1 The sequence and the role of the four phases that compose the eukaryotic cell cycle

The cell cycle is the process that every eukaryotic cell follows in order to proliferate, i.e. in order to duplicate its DNA and be divided into two daughter cells. This proliferation is achieved via two major phases, the $S$ phase and the $M$ phase, respectively. These phases are separated by two gap phases, the $G_1$ phase and the $G_2$ phase, which ensure the necessary cell growth, in order for the cell to duplicate its DNA and be divided. In this context, the cell cycle is reduced to the ordered sequence of these four phases, $G_1SG_2M$, which is summarized in **Figure 2.3**.

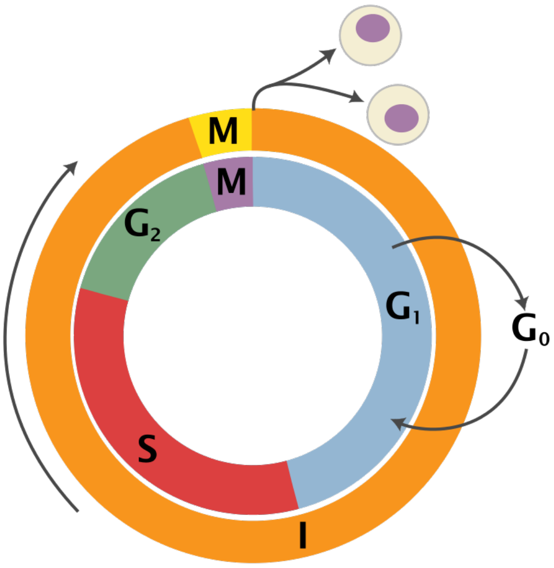


**Figure 2.3:** The four phases that compose the eukaryotic cell cycle [25]. The DNA of the cell is replicated in the $S$ phase, while the division of the cell in two clone daughter cells takes place in the $M$ phase. The cell cycle starts at the $G_1$ phase, which is responsible for the growth of the cell and its preparation for the DNA replication. After the DNA replication of the $S$ phase, the $G_2$ phase follows, in order to prepare the cell for its division during the $M$ phase. After the division of the $M$ phase, the two resulting daughter cells enter the $G_1$ phase and commence their own cell cycle. If the available nutrients and/or the cell size are not adequate, the cell cycle is arrested in the $G_1/S$ checkpoint (i.e. the restriction point) and the cell exits the cell cycle and enters the quiescent $G_0$ phase, in which it maintains its metabolic activities, but does not proliferate [26].

The first three phases of the cell cycle, $G_1$, $S$ and $G_2$, constitute the **interphase** (shown with orange color in **Figure 2.3**), the part of the cell cycle that is responsible for the growth and adaptation of the cell in preparation for its division. Typically interphase lasts for at least 91% of the total time required for the cell cycle.

The $\boldsymbol{G_1}$ **phase** (First growth phase or Post mitotic gap phase) is the phase that follows the $M$ phase and precedes the $S$ phase. It is very active in the biochemical level, since it is characterized by a high rate of biosynthetic activities that increase the supply of proteins and the number of organelles in the cell and cause its overall growth in size. This is the phase that prepares the cell for its DNA replication in the

$S$ phase, through the acquisition of the required energy and macromolecules. The duration of the $G_1$ phase is variable, even concerning cells of the same species [27].

The ***S* phase** (DNA synthesis) is the phase that follows the $G_1$ phase. This is the phase in which the DNA is synthesized via the replication of its chromosomes. After the completion of the DNA synthesis, each chromosome consists of two sister chromatids and the amount of the DNA in the cell has doubled, but the ploidy and number of chromosomes are unchanged. The $S$ phase is characterized by significantly lower biochemical and biosynthetic activity compared to the $G_1$ phase, since it mainly utilizes the tools and energy that were obtained in the $G_1$ phase. This means that RNA trasncription and protein synthesis are rare in the $S$ phase. An exception is histone proteins, which are produced and utilized in the $S$ phase for the packing and ordering of the DNA [28, 29, 30].

The **$G_2$ phase** (Second growth phase or Pro mitotic gap phase) is the phase that follows the $S$ phase and precedes the $M$ phase. Its role is to prepare the cell for its division in the $M$ phase, mainly via the reorganization of the cell's microtubules in a spindle form. This process, as well as the cell division to follow in the $M$ phase, require a special set of proteins and macromolecules. As a result, the $G_2$ phase is characterized by a high rate of biochemical and biosynthetic activity, similar to the $G_1$ phase.

The interphase is followed by the **mitotic phase**, which includes the events that lead to the division of the cell, which has passed through the interphase, into two identical daughter cells (clones). The mitotic phase is a very brief part of the cell cycle and consists only of the $M$ phase.

The ***M* phase** (Mitosis) is the phase that follows the interphase. It is a brief and highly regulated set of events that lead to the nuclear division (karyokinesis) through the separation of the chromosomes of the cell into two identical sets and finally to the division of the cell into two identical cells (cytokinesis) that equally share the nuclei, cytoplasm, organelles and cell membrane of the parent cell [31]. The steps that encompass the equal sharing of the contents of the parent cell and lead to its division into two identical daughter cells are summarized in **Figure 2.4**.

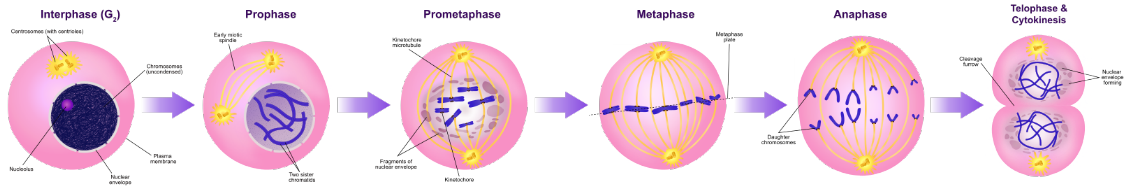


**Figure 2.4:** The distinct steps of the mitotic phase, which distribute the contents of the parent cell to two equal shares and lead to its physical division to two idential daughter cells [32].

### 2.2.2 The dynamic transcriptional network of macromolecules that regulates the evolution of the eukaryotic cell cycle

The cell cycle is a regulated sequence of events that realizes cell proliferation through the duplication of the parent cell's DNA, which is followed by the division of the parent cell into two identical daughter cells. The regulation of the ordered events of the phases of the cell cycle is ensured by a set of macromolecules within the context of a dynamically evolving transcriptional network [33, 34, 35].

The major regulators are couples of cyclins and cyclin-dependent kinases (CDKs). Cyclins are proteins that are synthesized at specific stages of the cell cycle. Their role is to activate the CDKs, which are constitutively expressed in cells. With their activation, the CDKs catalyze a common biochemical reaction called phosphorylation that activates or inactivates target proteins to orchestrate coordinated entry into the next phase of the cell cycle [36, 37]. In the case of the $G_1$ phase the cyclin-CDK complexes also determine the timing of the phase events.

The regulation of the cell cycle is also governed by a number of endogenous inhibitors that terminate the cell cycle in the case something goes wrong, e.g. in the case of genetic damage or uncontrolled division. These inhibitors consist of two major families of genes, the *cip/kip* (CDK interacting protein/Kinase inhibitory protein) family and the $INK4a/ARF$ (Inhibitor of Kinase 4/Alternative Reading Frame) family. The *cip/kip* family includes the genes $p21$, $p27$ and $p57$. They halt the cell cycle in $G_1$ phase by binding to and deactivating cyclin-CDK complexes. The $INK4a/ARF$ family includes $p16^{INK4a}$, which binds to $CDK4$ and arrests the cell cycle in $G_1$ phase, and $p14^{ARF}$ which prevents $p53$ degradation. The cell cycle inhibitors regulate the cell cycle in the case that the sequence of processes of its four distinct phases is not being realized properly, which aids the prevention and the management of cancerous tissue development. In that respect, these endogenous inhibitors are also characterized as tumor supressors and there have been attempts to replicate them synthetically. For instance, synthetic inhibitors for the $Cdk4/6$ activities, which are prevalent in human cancers, have been developed, with promising results in treating breast cancer [38, 39].

The effect that the couples of cyclins and CDKs have on the regulation and evolution of the cell cycle is demonstrated in **Figure 2.5**. The CDKs are present inside the cell at all times, while the cyclins are dynamically produced during the progression of the cell cycle. Depending on the dynamic state of the transcriptional network, which includes both internal and extracellular signals, the cyclin production and presence inside the cell changes also dynamically. The cyclins footprint inside the cell defines the existing cyclin-CDK pairs, which regulate the events of the current phase of the cycle and drive the progression to the following phase. Moreover, the presence of endogenous and synthetic inhibitors halts the evolution of the cell cycle in the case that its progression is defective, mainly by bonding with the CDKs that exist in the cells and deactivatng the cyclin-CDK complexes. Cyclins and inhibitors have opposite roles, in that the cyclins regulate the progress of the cell cycle, while the inhibitors regulate the suspension of the cell cycle. The balance between the effects of cyclins and inhibitors comes in the form of the internal and extracellular signals of the transcriptional network.

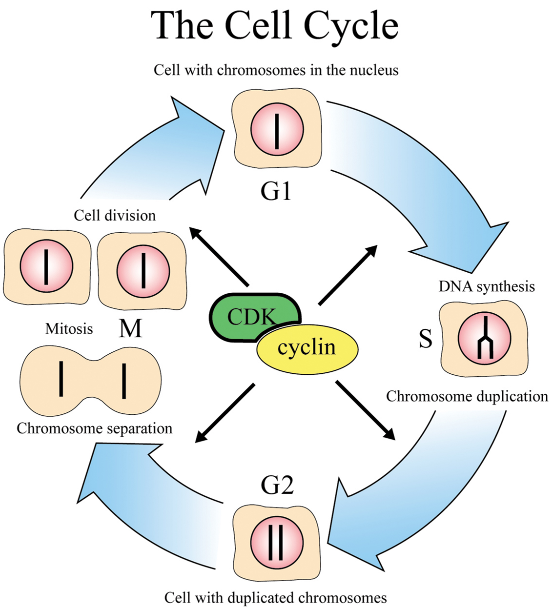


**Figure 2.5:** The regulation of the eukaryotic cell cycle by couples of cyclins and CDKs [40]. The couples of cyclins and CDKs drive the cell cycle to progress through the consecutive phases. The CDKs already exist in the cell, while the cyclins are produced according to the current phase and the existing conditions. If the cell cycle evolution becomes defective, the endogenous and synthetic inhibitors are bonded to the CDKs deactivating the cyclin-CDK pairs and halting the cell cycle progression. The countervailing effects of cyclins and inhibitors are regulated by a dynamic transcriptional network, which employs both internal and extracellular signals [26].

### 2.2.3 The major chekpoints of the eukaryotic cell cycle and the resting $G_0$ phase

As mentioned in the previous paragraph, the evolution of the cell cycle progression is highly regulated, using a set of macromolecules (cyclins, CDKs, inhibitors) within the context of a dynamic transcriptional network of signals. The regulation of the cell cycle evolution also encompasses a set of checkpoints, i.e. a set of standards and requirements that are inherent to specific points in its progression, in order to ensure that its events occur properly and within the right schedule.

The first major cell cycle chekpoint is the $G_1/S$ checkpoint and refers to the $G_1$ phase. More specifically, according to the extracellular growth factor and if the nutrients are not adequate and the cell is malnourished, then it is considered to be unfit for proliferation. As a result, it leaves the cell cycle and enters the quiescent $G_0$ phase (see **Figure 2.3**). The cells that reside in the **$G_0$ phase** stay metabolically active, but they do not grow or proliferate and their rates of the biochemical processes and protein synthesis are reduced. If extracellular signals and suitable growth factors apply, the cell may reenter the cell cycle and start proliferating.

The cells that pass the $G_1/S$ checkpoint progress to the $S$ phase and are committed to performing cell division. This is why the $G_1/S$ checkpoint is also called restriction point, since it determines which cells will be restricted to the quiescent $G_0$ phase and will not perform cell division.

The next major cell cycle checkpoint is the $G_2/M$ checkpoint, which refers to the $G_2$ phase. More specifically, it ensures that the cell is equiped with enough cytoplasm and phospholipids for the two daughter cells that will result from the mitosis. Once the cell has reached this checkpoint, it is already committed to undergoing mitosis and, contrary to the $G_1/S$ restriction checkpoint, the $G_2/M$ checkpoint acts as a regulator for the timing that the cell is ready to enter the mitotic phase, rather than a judge for whether the cell will finally divide or not.

In the same way as the $G_2/M$ checkpoint, the final major cell cycle checkpoint, the metaphase checkpoint, is responsible for scheduling the timing of the events of the cell cycle, in that it ensures that the spindle has formed and that all of the chromosomes are aligned at the spindle equator before anaphase begins.

The proper evolution of the cell cycle requires that all the checkpoints are taken into consideration and are satisfied before the progression of the events of the cycle for each cell. However, this is not always the case. Defects in the complex network of macromolecules and signals that regulate the cell cycle progression may cause the cells to speed from one phase to the next with no consideration for the aforementioned checkpoints. This causes the accumulation of mutations in the cells and the manifestation of cancerous tissue, which, due to this property of speeding and even skipping through phases, often grows rapidly.

## 2.3 The Oncosimulator

### 2.3.1 The outline of the clinical environment and workflow within which the Oncosimulator is utilized as a discrete top-down approach for cancer modeling

The Oncosimulator is a typical example of an application for personalized medicine in cancer. It is designed to receive as input patient-specific medical data (imaging, histopathological, etc.), to perform a series of simulations on the evolution of the neoplasm and finally to produce the results of the simulations in the form of the growth percentage of the tumor, as well as the change of its composition concerning the various cell categories, e.g. proliferative, dormant, differentiated. As such, it is by definition built for clinical practice, to be used as a clinical assistant and advisor to medical practicioners. However, its nature as a cancer evolution simulator also renders it an intersting tool for medical training and research.

The Oncosimulator comes in different versions that are customized for the different types of cancer, e.g. lung cancer, breast cancer, glioblastoma, nephroblastoma. However, each version of the Oncosimulator has a similar structure and is used in the context of a common workflow, as shown in **Figure 2.6**.

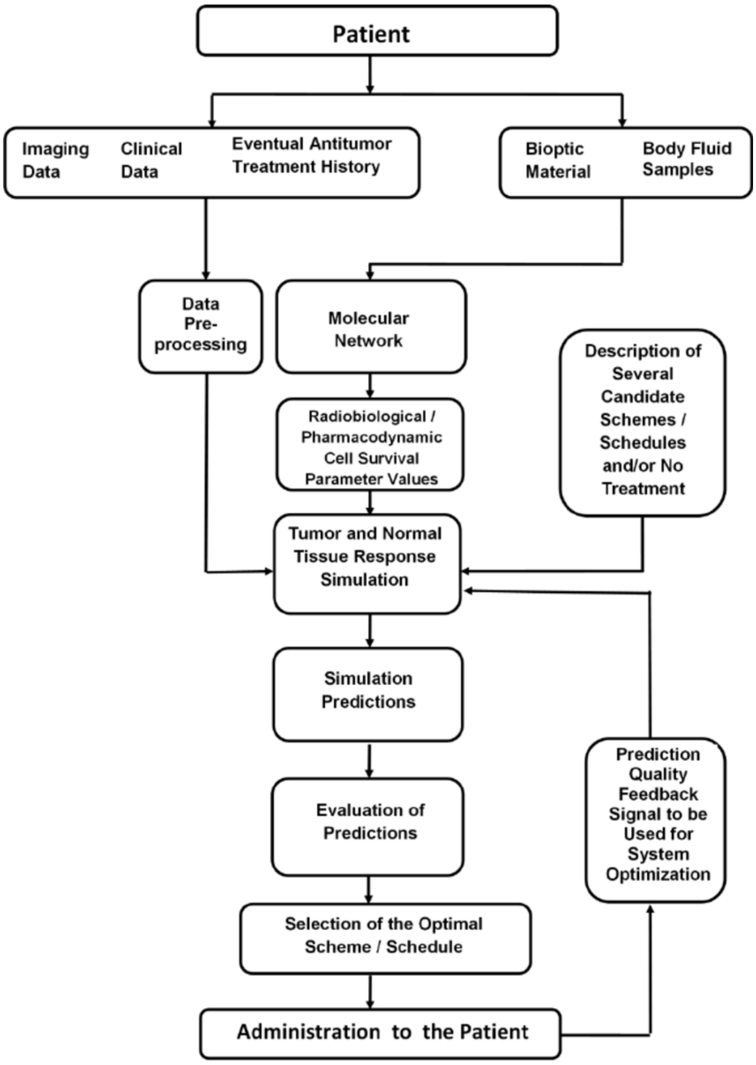

Patient
Imaging Data
Clinical Data
Eventual Antitumor Treatment History
Bioptic Material
Body Fluid Samples
Data Pre-processing
Molecular Network
Radiobiological / Pharmacodynamic Cell Survival Parameter Values
Description of Several Candidate Schemes / Schedules and/or No Treatment
Tumor and Normal Tissue Response Simulation
Simulation Predictions
Evaluation of Predictions
Prediction Quality Feedback Signal to be Used for System Optimization
Selection of the Optimal Scheme / Schedule
Administration to the Patient

**Figure 2.6:** The diagram that describes the workflow within which the Oncosimulator is employed as a clinical assistant and advisor [41]. The patient specific medical data, containing imaging data, clinical data and data concerning the previous treatment history, are preprocessed and given as direct input to the simulation. Moreover, the biopsy and body fluid data pass through a molecular network and a radiobiological/pharmacodynamic cell survival parameter generator, providing the simulation with the parameter values that describe the simulated biological processes. Finally, the simulation is provided with the treatment schedule to be simulated (the simulation of the free growth of the tumor is also an option). The results of the simulation are expressed as the volume growth/shrinkage percentage of the tumor and as its final composition, with concern to the percentage of each cell population, e.g. proliferative, differentiated, dead, dormant. The medical doctor assesses the results of the simulation and chooses the optimal treatment plan for the patient. After the treatment has been completed and its outcomes recorded, they are compared to the ones that the simulation predicted. This comparison is utilized in a closed feedback loop that is responsible for the optimization of the simulation, so that it approximates better the phenomena that it represents in silico.

As a first step, the patient-specific biomedical data are collected. These involve the clinical data and the previous treatment history of the patient, the imaging data, the body fluid samples and the biopsy material. These data are then given as input to the model, along with the treatment modality to be simulated. The body fluid samples and the biopsy material data pass first through a molecular network and a radiobiological/pharmacodynamic cell survival parameter generator. After the model is provided with the raw biomedical data of the patient and the parameters that are derived from them, as well as the therapeutic scheme to be simulated (notice how no treatment can also be simulated in the so called free growth mode simulation), the simulation is performed and then its results are produced in the form of predictions. These predictions include the growth percentage of the tumor, as well as its final composition concerning the various cell categories. After, the predictions are evaluated by the medical doctor and the optimal therapeutic scheme is applied to the patient. Finally, the results of the actual therapy applied to the patient are given as feedback to the model, where they can be compared to the model's predictions and urge the process of the model optimization via the adjustment of the granularity of the model in order for it to better capture the reality of the biomedical processes at play.

Observing **Figure 2.6**, which shows the common workflow that all the versions of the Oncosimulator follow, it becomes clear that the Oncosimulator is a data driven model. As such, it is a typical example of the top-down modeling approach that integrates clinical observation with scientific medical knowledge (see **Chapter 2.1.2**). Indeed, the Oncosimulator combines the clinical observation in the form of the wealth of patient medical data with the existing biomedical knowledge that is implemented in hypomodel components (e.g. the molecular network) and is dynamically adjusted in order to better fit reality via the continuous optimization feedback loop, as is typically done in the top-down approach.

As a model that follows the top-down approach, the Oncosimulator implements the core biological system that it models, i.e. the cancerous neoplasm and its microenvironment of normal and necrotic cells, starting from the highest spatial scale and moving to the lowest, using the summarize and jump strategy. In a first primary level, this is achieved by superimposing a 3D mesh of discrete voxels on the area of interest, i.e. on the tumor. Each voxel represents a cell, which ideally corresponds to a biological cell, though in reality the actual biological cells are way more than can be practically implemented in a data structure and simulated in silico. This means that the voxels of the 3D mesh actually correspond to geometrical cells, which are filled with a number of biological cells that form equivalence classes, in which cells are simulated as one.

The cells are simulated via the simulation of their transitions between the different phases inside and out of the cell cycle. These transitions are implemented in dicrete time steps. In this context, the Oncosimulator can be characterized as a discrete entity-discrete event model, which falls under the broad category of dicrete models. After the transitions between the various cell phases is completed, the results of the simulation in the cellular scale, which involve cell extermination and generation, among others, are "summarized" and then a "jump" is made into the tissue level, where biomechanical simulations are performed for the adjustment of the volume of the tumor.

Even though the Oncosimulator is a primarily discrete model, it also has auxiliary continuous and stochastic modules. For example, the continuous method is introduced in the form of an equation that expresses the condition for the free growth of the tumor, while the stochastic method is introduced for the selection of a random direction for the shrinkage or the growth of the 3D mesh tumor.

### 2.3.2 The "summarize and jump" Oncosimulator algorithm, which represents the solid tumor as a 3D matrix and simulates its evolution in discrete time steps

As described in the previous paragraph, the Oncosimulator represents the solid tumor in silico as a 3D matrix. Each item of the 3D matrix corresponds to a geometrical cell, i.e. to a voxel in the medical image of the cancerous region of interest that contains the solid tumor. Moreover, the simulation is performed in the 4D space, since the tumor evolution is simulated in discrete time steps, which characterizes the Oncosimulator as a discrete entity-dicrete event model. The "summarize and jump" algorithm, which was briefly outlined in the previous paragraph, is described in full detail in the flowchart of **Figure 2.7**.

Observing **Figure 2.7**, it becomes clear that the algorithm for the Oncosimulator consists of seven consecutive scans of the 3D mesh of geometrical cells that covers the area of interest, i.e. the tumor and its normal and necrotic microenvironment.

The **first scan (Tumor_initialization)** takes care of the initialization of the tumor in silico in the form of a 3D mesh of geometrical cells. This initialization is based on the patient-specific input data and the input parameters of the model. Notice how these inputs may lead to an unacceptable tumor, if the condition for free growth (**Equation 2.1**) is not met, in which case the simulation is terminated and no results are produced. The variables that are present in **Equation 2.1** form a subset of the parameters of the simulation, which are presented in detail later on (**Chapter 2.3.4**). If the initialization of the 3D mesh of cells is completed successfully, the geometrical cells of the 3D mesh and their variables are set for the start of the simulation, which commences from time $t = 0$. This first scan is performed only once and is not repeated for any time $t > 0$.

$$(1 + P_{sym}) \cdot (1 - P_{sleep} + P_{sleep} \cdot \frac{P_{G0toG1}/T_{G0}}{R_A + 1/T_{G0}}) \cdot e^{-R_A \cdot T_C} \geq 1 \qquad (2.1)$$

Then follows the **second scan (Therapy)**, which is responsible for the computation of the percentage of the cells that are affected by therapy, as well as the implementations of these effects of therapy on all the cells of the 3D mesh. The effects of therapy roughly consist of a number of biological cells inside the geometrical cells of the 3D mesh being exterminated and, as a result, the population of the cells that reside in the various cell phases inside the equivalence classes being adjusted appropriately, along with the other variables that accompany them, e.g. the time that the cell is destined to reside in the current cell phase. This scan is applied only for the time points that fall in the time duration that the therapeutic scheme is actively applied, i.e. for time $0 < t < t_{last\_admin}$, where $t_{last\_admin}$ is the time point when the last administration of therapy is applied.

The second scan is followed by the **third scan (Growth)**, which is responsible for the transitions that happen physiologically and dynamically between the cell phases for the biological cells that are organized in equivalence classes in the geometrical cells of the 3D mesh. First, spontaneous apoptosis is implemented similar to the way that the effects of therapy are implemented in the second scan. The percentage of the cells to apoptose is computed and then the apoptosis is implemented via the adjustment of the populations and the other variables of the cells that reside in the various states or equivalence classes. Then, the cell cycle transitions are applied for the cells that have completed their time in their current phase and, again, the populations and the times are redefined.

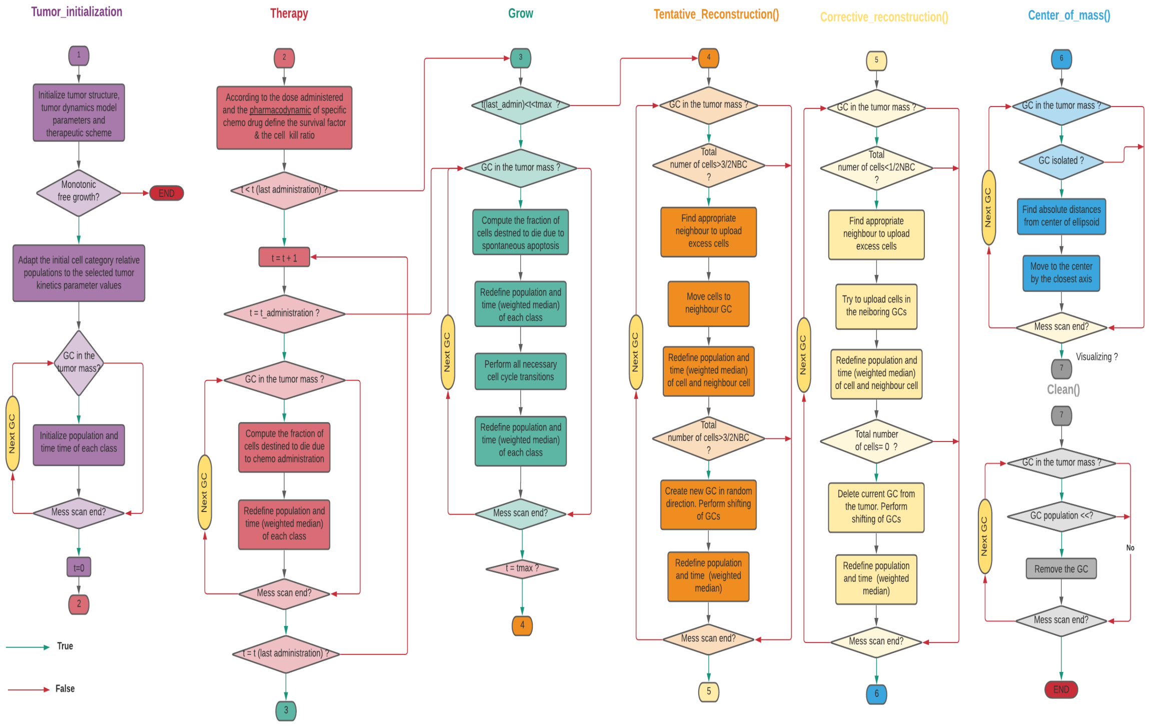

Tumor_initialization
1
Initialize tumor structure, tumor dynamics model parameters and therapeutic scheme
Monotonic free growth?
END
Adapt the initial cell category relative populations to the selected tumor kinetics parameter values
GC in the tumor mass?
Next GC
Initialize population and time time of each class
Mess scan end?
t=0
2
True
False
Therapy
2
According to the dose administered and the pharmacodynamic of specific chemo drug define the survival factor & the cell kill ratio
t < t (last administration) ?
t = t + 1
t = t_administration ?
GC in the tumor mass ?
Next GC
Compute the fraction of cells destined to die due to chemo administration
Redefine population and time (weighted median) of each class
Mess scan end?
t = t (last administration) ?
3
Grow
3
t(last_admin)<t<tmax ?
GC in the tumor mass ?
Next GC
Compute the fraction of cells destned to die due to spontaneous apoptosis
Redefine population and time (weighted median) of each class
Perform all necessary cell cycle transitions
Redefine population and time (weighted median) of each class
Mess scan end?
t = tmax ?
4
Tentative_Reconstruction()
4
GC in the tumor mass ?
Total numer of cells>3/2NBC ?
Next GC
Find appropriate neighbour to upload excess cells
Move cells to neighbour GC
Redefine population and time (weighted median) of cell and neighbour cell
Total number of cells>3/2NBC ?
Create new GC in random direction. Perform shifting of GCs
Redefine population and time (weighted median)
Mess scan end?
5
Corrective_reconstruction()
5
GC in the tumor mass ?
Total numer of cells<1/2NBC ?
Next GC
Find appropriate neighbour to upload excess cells
Try to upload cells in the neiboring GCs
Redefine population and time (weighted median) of cell and neighbour cell
Total number of cells= 0 ?
Delete current GC from the tumor. Perform shifting of GCs
Redefine population and time (weighted median)
Mess scan end?
6
Center_of_mass()
6
GC in the tumor mass ?
GC isolated ?
Next GC
Find absolute distances from center of ellipsoid
Move to the center by the closest axis
Mess scan end?
Visualizing ?
7
Clean()
7
GC in the tumor mass ?
GC population <<?
Next GC
No
Remove the GC
Mess scan end?
END

**Figure 2.7:** The flowchart of the "summarize and jump" Oncosimulator algorithm. The solid tumor is represented in silico as a 3D matrix of geometrical cells, each one containing a multitude of biological cells. Moreover, the evolution of the tumor is simulated in discrete time steps (4D simulation), which characterizes the Oncosimulator as a discrete entity-discrete event model. The 3D matrix is processed in seven consecutive scans for each discrete time step. The first scan is responsible for the initialization of the 3D matrix according to the provided medical imaging data (applied only once at the start of the simulation). If the free growth condition is not satisfied (see **Equation 2.1**), the simulation is terminated prematurely. The second scan is responsible for the simulation of the effects of therapy and the third scan is responsible for the simulation of the cell cycle transitions. Combined they implement the cytokinetic diagram of **Figure 2.8** and complete the simulation in the cellular level. With the simulation "summarized" in the cellular level, a "jump" is made in the tissue level, where the fourth, fifth and sixth scan collaborate for the completion of the biomechanical part of the simulation. More specifically, the fourth scan is responsible for managing the excess of biological cells inside the geometrical cells and growing the tumor, the fifth scan is responsible for managing the shortage of biological cells inside the geometrical cells and shrinking the tumor and the sixth scan is responsible for maintaining the solid shape of the tumor tissue in the 3D matrix. The process is repeated for each time step and for the second up to the sixth scan, the simulation continuously "jumping" from cellular to tissue level and vice versa. Finally, the seventh scan categorizes the geometrical cells with too few biological cells as necrotic (useful for the 3D image reconstruction of the tumor).

The combination of the second with the third scan implements the simulation in the cellular level, i.e. the transitions between the different cell phases for each equivalence class in the geometrical cells of the 3D mesh of the tumor (third scan), as well as the effects that the chosen therapeutic scheme has on these transitions and the composition of the tumor in general (second scan). The part of the simulation that is implemented via these two scans is summarized in the cytokinetic diagram of **Figure 2.8**.

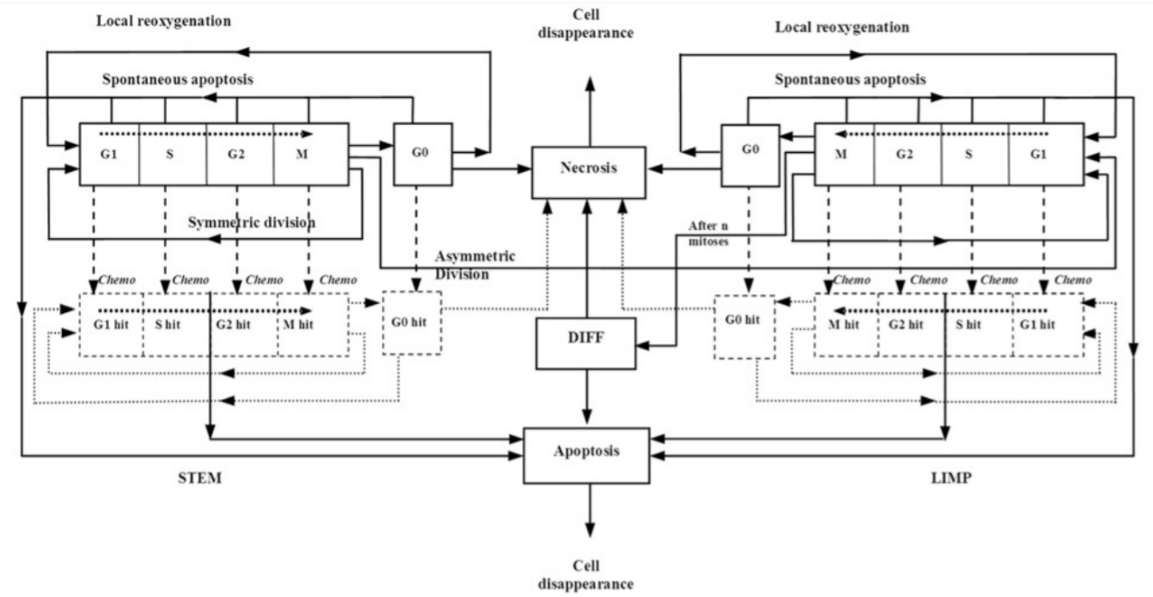


**Figure 2.8:** The cytokinetic diagram that visualizes the transitions between the phases inside and out of the cell cycle and the effects of therapy [42]. The cells transition consecutively between the $G_1$, $S$, $G_2$ and $M$ phases, inside the cell cycle. A portion of the cells that are produced in the $M$ phase reenter the cell cycle in the $G_1$ phase, while the rest leave the cell cycle and enter the quiescent $G_0$ phase. Some of the cells that reside in the $G_0$ phase reenter the cell cycle in the $G_1$ phase, while some die via necrosis. If the division of the $M$ phase is symmetric, its products are STEM cells (left side of the diagram), while if it is asymmetric LIMP cells are produced (right side of the diagram), which differ in that they can repeat the cell cycle a limited amount of time $n$, before they are differentiated terminally. A portion of cells, no matter if STEM or LIMP and no matter the phase inside the cell cycle, disappear via spontaneous apoptosis. The cells that are hit by therapy perform one more cell cycle before they apoptose.

A first observation of **Figure 2.8** demonstrates the cycling of the cells through the various cell phases inside the cell cycle, starting from the first gap phase $G_1$, going on to the DNA synthesis phase $S$, then to the second gap phase $G_2$ and finally to the mitotic phase $M$. From the mitotic phase, two daughter cells are produced, which have the option of going on to phase $G_1$ and starting the cell cycle from scratch or going into the resting $G_0$ phase, out of the cell cycle. In the first case, the newborn cell immediately starts the cell cycle, while in the second case it goes into a quiescent state, e.g. due to a lack of nutrients like oxygen, where it keeps serving its purpose, but not proliferating, until it either dies via necrosis or reenters the cell cycle from the $G_1$ phase, e.g. due to reoxygenation. Moreover, at any time and in any cell phase, inside or out of the cell cycle, a percentage of cells is going through spontaneous apoptosis, which, in the long run, leads to cell disappearance, same as necrosis.

A similar route is followed by the cells that are hit by therapy, e.g. chemotherapy, with the difference that they first pass through an intermediate stage, where they keep transitioning through the cell phases and can even die via necrosis, e.g. in the case that they reside in the $G_0$ phase. In that way, the delay of the effects of therapy is included in the simulation and, as a result, a more realistic model for therapy is acquired.

One last very important feature of the cytokinetic diagram is the fact that it models the behavior of two distinct big categories of cells, the STEM cells and the LIMP cells. The STEM cells, which are modeled with reference to the left side of the diagram, are the cells that keep proliferating following the various mechanisms as they were described before, while the LIMP cells, which are modeled with reference to the right side of the diagram, are the cells that have limited mitotic potential (LImited Mitotic Potential), which means that they also proliferate and follow the mechanisms that were presented before, but only for a limited amount of time. A LIMP cell is produced as the effluent of the asymmetric division of a STEM cell during mitosis, while a symmetric division causes two STEM daughter cells. The model for a LIMP cell (right side of the diagram) is a mirror of the model for a STEM cell (left side of the diagram), with the only difference that after $n$ repetitions of the cell cycle the LIMP cell is terminally differentiated and finally disappears either via necrosis or via apoptosis.

The implementation of the cytokinetic diagram of **Figure 2.8** via the second scan and the third scan of the Oncosimulator algorithm ensures the simulation of the events and bioprocesses that take place in the cellular scale in the form of the cell cycle, the transitions between the various cell phases, inside and out of the cell cycle, the disappearance of the cells via apoptosis, necrosis and differentiation, as well as the manner in which cancer therapy interacts with these processes, as they were presented in previous chapters. The evolution of these processes and the transition of the cytokinetic diagram are dictated by a set of parameters, which are part of the input of the model. These parameters are presented in detail in the dedicated chapter (**Chapter 2.3.4**), where they can be put in better context.

With the simulation in the cellular scale "summarized" via the redefinition of the populations of cells in the equivalence classes in the geometrical cells of the 3D mesh tumor, along with other variables, such as the time that remains before the cell equivalence class transitions to another phase, a "jump" is made in the tissue scale, where the biomechanical simulation takes place, for the spatial shrinkage or the expansion of the tumor. This biomechanical simulation is performed via the next three scans, the fourth scan, the fifth scan and the sixth scan.

The **fourth scan (Tentative_Reconstruction)** is responsible for the expansion of the tumor. More specifically, it scans the 3D mesh and for the geometrical cells that contain an excess of biological cells, first an attempt is made to unload some on neighbor cells and then, in case the excess is still present, a new geometrical cell is created, in order to sustain the extra biological cells. In that way, the accumulated excess of cells that was acquired via the simulation on the cellular scale is equally distributed to the entire tumor and ultimately leads to the growth of the tumor and its spatial expansion. The excess of biological cells in a gemotetrical cell is considered as the concentration of more than $\frac{3}{2} \cdot NBC$ biological cells in one geometrical cell, where $NBC$ is the default number of biological cells per geometrical cell.

The **fifth scan (Corrective_Reconstruction)** is responsible for the shrinkage of the tumor. More specifically, it scans the 3D mesh and for the geometrical cells that have a lack of biological cells, first an attempt is made to unload them on the neighbor geometrical cells and then, if there are no biological cells remaining in the geometrical cell, the geometrical cell is removed from the 3D mesh tumor. In that way the accumulation of lack of cells that is the result of the simulation on the cellular scale is equally distributed to the entire tumor and ultimately leads to the reduction of the tumor and its spatial shrinkage. The lack of biological cells in a gemotetrical cell is considered as the concentration of less than $\frac{1}{2} \cdot NBC$ biological cells in one geometrical cell, where $NBC$ is the default number of biological cells per geometrical cell.

Finally, the **sixth scan (Center_of_mass)** completes the biomechanical simulation on the tissue scale by making sure that the creation and destruction of geometrical cells that is the result of the fourth scan and the fifth scan, respectively, does not leave the tumor with an unnatural geometry, e.g. with isolated geometrical cells. This is ensured by discovering the isolated geometrical cells of the 3D mesh, calculating their distance from the center of mass of the tumor and moving them toward it by the closest axis until they are no more isolated.

Thus, the biomechanical part of the simulation is completed via the fourth scan, the fifth scan and the sixth scan of the Oncosimulator algorithm. Notice that the biomechanical simulation contains stochastic processes for the generation of the direction in which the target biological cells will be unloaded, as well as the direction that the tumor will grow or shrink.

With the completion of the biomechanical part of the simulation at the tissue scale and its results "summarized" in the form of the distribution of the biological cells on the entire 3D mesh of geometrical cells, as well as the appropriate expansion or shrinkage of the 3D mesh tumor, a "jump" is made back to the cellular scale to start the scans again from the second scan and for the next time step ($t = t + 1$). The scans of the 3D mesh of geometrical cells, from the second scan up to the sixth scan, are repeated for the consecutive time steps, until $t = t_{max}$, i.e. until the time point set as the end of the simulation. That way, a 4D simulation of the evolution of the tumor and its response to therapy is performed, in the 3D geometrical space and in the 1D temporal space.

Finally, the simulation is completed with the **seventh scan (Clean)**, which cleans the geometrical cells which contain too few biological cells off the tumor by characterizing them as necrotic tissue. This is an important step for the reconstruction of the tumor, especially when it is being illustrated in image.

### 2.3.3 The hypermatrix notation that is employed for the implementation of the Oncosimulator algorithm

The hypermatrix notation was introduced in the context of the Oncosimulator, in order to implement the algorithm that was presented previously. This is achieved by translating the tumor to a 3D mesh data structure with voxels that correspond to geometrical cells that contain a number of biological cells along with a set of variables that define the transitions of the cells between the various cell phases and states that are summarized in the cytokinetic diagram of **Figure 2.8** [23].

From the mathematical and algorithmic data structure point of view, the hypermatrix $\underline{\overline{a}}$ is a matrix of [a matrix of [... of [matrices (or vectors)]...]] that is created by the superposition of a discretization mesh on the anatomic region of interest, i.e. the cancerous neoplasm and its necrotic and normal tissue microenvironment, using the patient-specific medical data and especially the medical imaging data of the patient (e.g. MRI, CT/PET). That way, with appropriate reconstruction techniques and using the rest of the patient biomedical data (e.g. fluid samples, histopathological data), the current state of the tumor can be represented in silico in a discrete 3D mesh of geometrical cells, i.e. voxels, that has a shape that fits the one of the specific imageable tumor. The hypermatrix $\underline{\overline{a}}$ also has a fourth dimension that corresponds to the temporal evolution of the tumor.

As has been mentioned already, a **geometrical cell** is basically a voxel of the 3D mesh of the hypermatrix that discretizes the area of interest, or the basic component of the 4D mesh of the hypermatrix, if the dimension of time is also considered. The geometrical cell, if it is adequately small, corresponds ideally to one sole biological cell. However, this is computation-wise impracticable as far as the in silico implementation is considered. This means that, practically, the geometrical cell contains a number of actual biological cells and that with the increase of the size of the geometrical cell, this number also increases, while with the decrease of the size of the geometrical cell, the number of biological cells it contains also decreases.

The biological cells are further organized in the so called **equivalence classes** inside the geometrical cells. An equivalence class consists of the biological cells that reside inside a geometrical cell and that are in the same phase inside or out of the cell cycle. In fact, the cells that form an equivalence class are simulated as one cell and carry out the transitions of the cytokinetic diagram as one. This means that they share a number of variables, e.g. the time that remains before they perform the next transition on the cytokinetic diagram. This also means that, with the number of the equivalence classes inside each geometrical cell constant, the more biological cells a geometrical cell contains, i.e. the bigger the geometrical cell is, the more biological cells reside in the same equivalence class, which naturally reduces the resolution of the simulation. More specifically, bigger geometrical cell means lighter computations but reduced resolution, while smaller geometrical cell means heavier computations but higher resolution. In theory, the simulation with the perfect resolution occurs when the geometrical cell is so small that the concepts of geometrical cell, equivalence class and biological cell are reduced to the same concept.

With concern to symbolism, $\underline{a}$ stands for the hypermatrix and an element of the hypermatrix is symbolized as follows:

$$\begin{aligned} \underline{a}(x_i, y_j, z_k, p_l, t_n) &= (g^{ijkln}, N_p^{ijkln}, t_p^{ijkln}, h_p^{ijkln}, \tilde{h}_p^{ijkln}) \\ \underline{a}(t_0) &= \underline{a_0} \quad \text{initial state of the tumor} \end{aligned} \tag{2.2}$$

where:

$$\begin{aligned}
x_i \in [x_{min}, x_{max}] &: \text{the x coordinate in the 3D mesh} \\
&\phantom{:}\ x_{min} \text{ and } x_{max} \text{ denote the boundaries on the x axis} \\
y_j \in [y_{min}, y_{max}] &: \text{the y coordinate in the 3D mesh} \\
&\phantom{:}\ y_{min} \text{ and } y_{max} \text{ denote the boundaries on the y axis} \\
z_k \in [z_{min}, z_{max}] &: \text{the z coordinate in the 3D mesh} \\
&\phantom{:}\ z_{min} \text{ and } z_{max} \text{ denote the boundaries on the z axis} \\
t_n \in [0, t_{max}] &: \text{the time point coordinate in the 3D mesh} \\
&\phantom{:}\ t_{max} \text{ denotes the last time point of the simulation} \\
p_l \in \{G_1, S, G_2, M, G_0, A, N, D\} &: \text{the cell phase of the element} \\
&\phantom{:}\ G_1,\ S,\ G_2,\ M,\ G_0 \text{ denote the known cell phases} \\
&\phantom{:}\ A \text{ denotes the apoptotic cell phase} \\
&\phantom{:}\ N \text{ denotes the necrotic cell phase} \\
&\phantom{:}\ D \text{ denotes the remnants of dead cells} \\
p &: \text{phase within or out of the cell cycle} \\
g \in \{s, \tilde{s}\} &: \text{oxygen and nutrient provision} \\
&\phantom{:}\ s \text{ stands for sufficient nutrient provision} \\
&\phantom{:}\ \tilde{s} \text{ stands for insufficient nutrient provision} \\
N_p \in N_0 &: \text{number of biological cells in phase p} \\
&\phantom{:}\ N_0 \text{ the set of non negative integers} \\
t_p \in [0, t_{p,max}] &: \text{mean time spent in phase p} \\
&\phantom{:}\ t_{p,max} \text{ max time spent in phase } p \\
h_p \in [0, N_p] &: \text{number of therapy hit cells residing in phase p} \\
\tilde{h}_p \in [0, N_p] &: \text{number of non therapy hit cells residing in phase p}
\end{aligned}$$

The spatial coordinates $x_i$, $y_j$ and $z_k$, along with the time point coordinate $t_n$ define one specific geometrical cell of the 3D mesh tumor. The definition of the remaining variable, the cell phase $p_l$, defines an equivalence class within this geometrical cell. Notice how the full set of these variables, $x_i$, $y_j$, $z_k$, $t_n$ and $p_l$, describe a fully defined element of the 3D mesh tumor, with known properties, $g$, $N_p$, $t_p$, $h_p$ and $\tilde{h}_p$, as indicated by the $ijkln$ exponent in **Equation 2.2**. This means that an equivalence class within a geometrical cell is a fully defined component element of the 3D mesh tumor.

The process of parsing the geometrical cells of the 3D mesh tumor and accessing the properties of the fully defined equivalence class components inside of them, is visualized in **Figure 2.9**. The set of $x_i$, $y_j$, $z_k$ and $t_n$ variables define the different geometrical cells, while the $p_l$ variable defines the equivalence class inside them. With the definition of all five variables, access to a fully defined element of the 3D mesh tumor and to its five properties, $g$, $N_p$, $t_p$, $h_p$ and $\tilde{h}_p$, is provided.

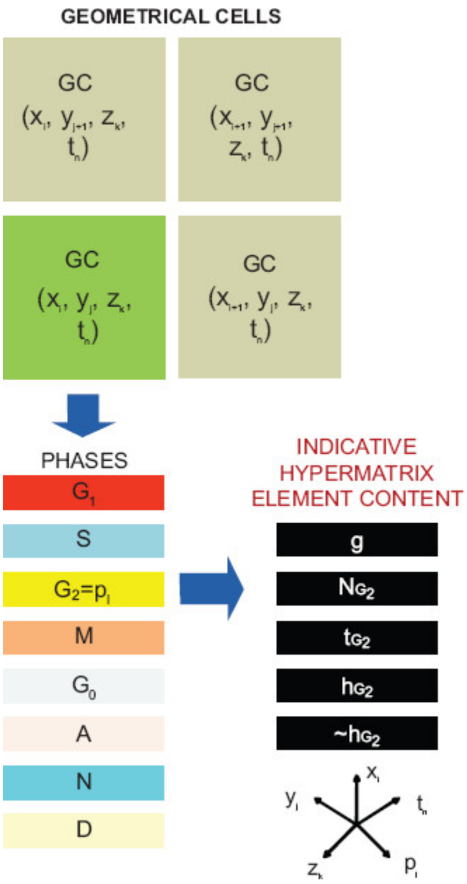

GEOMETRICAL CELLS
GC (x_i, y_{j+1}, z_k, t_n)
GC (x_{i+1}, y_{j+1}, z_k, t_n)
GC (x_i, y_j, z_k, t_n)
GC (x_{i+1}, y_j, z_k, t_n)
PHASES
G1
S
G2=p_l
M
G0
A
N
D
INDICATIVE HYPERMATRIX ELEMENT CONTENT
g
NG2
tG2
hG2
~hG2
x_i
y_j
t_n
z_k
p_l

**Figure 2.9:** The hypermatrix data structure organized in geometrical cells and equivalence classes [23]. The set of $x_i$, $y_j$, $z_k$ and $t_n$ variables (spatiotemporal coordinates) define the different geometrical cells. The $p_l$ variable (phase inside or out of the cell cycle) further defines an equivalence class inside each geometrical cell. Each equivalence class inside a specific geometrical cell, specified by the values for $x_i$, $y_j$, $z_k$, $t_n$ and $p_l$, is characterized by the values for $g$, $N_p$, $t_p$, $h_p$ and $\tilde{h}_p$ (see **Equation 2.2**).

The simulation is performed by accessing and modifying the hypermatrix in discrete time steps. The evolution of the hypermatrix in time is achieved via the application of an operator $f$ on it for each discrete time step, as expressed in **Equation 2.3**.

$$\overline{a}(t_{n+1}) = f(\overline{a}(t_n)) \tag{2.3}$$

The operator $f$ is applied on all the element of the hypermatrix for each discrete time step. It summarizes all the bioprocesses that happen dynamically and that have an effect on the evolution of the tumor. In that way, the simulation is performed properly and the appropriate results are accomplished. In order for the operator $f$ to contain the information for all the bioprocesses that happen in each time step, it is decomposed into a series of partial operators that are applied sequentially, as shown in **Equation 2.4**. The operators are applied with order from right to left. The order in which the partial operators are applied is crucial for the outcome of the simulation, since the operators are not commutative (i.e. interchangeable).

$$f = f^U f^E f^C f^H f^O f^T \tag{2.4}$$

The first partial operator that is applied is the **operator $\boldsymbol{f^T}$**. This is the operator that increases the time by one time step. The typical size of a time step is $1h$, the approximate duration of mitosis, the shortest cell phase.

The next partial operator that is applied is the **operator $\boldsymbol{f^O}$**. This operator takes into account the nutrient and oxygen provision status of the geometrical cells, based primarily on the medical imaging data, in order to determine the necrotic, normal and proliferative regions of the anatomical area of interest and, going into further detail, to approximate the distribution of the cells in the various cell phases, inside or out of the cell cycle, for the various regions of the anatomical area of interest.

Then follows the application of the **operator $\boldsymbol{f^H}$**, which is responsible for the implementation of the effects of therapy. The effect of this operator depends on the therapeutic scheme that is simulated. However, this is the operator that incorporates any eventual molecular perturbators of the cell surviving fraction and its results are typically summarized in the form of cell survival ratios.

After the operator $f^H$, the **operator $\boldsymbol{f^C}$** is applied. This is the operator that implements the physiological bioprocesses and events that take place in the lifespan of a cell, i.e. the cell cycle and cell deactivation or death through the resting $G_0$ phase, necrosis, apoptosis or differentiation. This operator basically implements the cytokinetic diagram of **Figure 2.8**. The transitions between the cell phases happen when the maximum time of the current phase has been fulfilled. These times are incorporated in the model as part of the input parameters, that are presented in **Chapter 2.3.4**.

Then follows the biomechanical simulation via the application of the **operator $\boldsymbol{f^E}$**. This operator is responsible for the spatial expansion or shrinkage of the tumor. It has been implemented using various techniques and methods. The naive method chooses a random direction from the 6 possible for the expansion or shrinkage of the tumor. This method causes a premature atypical fragmentation of the tumor in the context of radiotherapeutic schemes. A more refined approach chooses the direction of expansion or shrinkage of the tumor based on the tumor's current shape, so that, e.g. in the case of shrinkage, the direction that is chosen is the one with the more geometrical cells inside the tumors. That way, the excessive fragmentation of the first approach is avoided.

Finally, the evolution of the tumor for one time step is completed with the application of the **operator $\boldsymbol{f^U}$**. This operator recalculates the nutrient and oxygen provision, after the biomechanical simulation, in order to take into account any eventual expansion or shrinkage of the tumor that would lead to a perturbation of the previous metabolic potential field.

This is an indicative only decomposition of the **operator *f*** into partial operators. This decomposition can be enriched with more partial operators and appropriately adjusted, in order for the simulation to better represent reality for the selected model, e.g. depending on the type of cancer and the treatment modality.

Notice how the partial operators correspond to the consecutive scans of the algorithm, as they were presented previously. This means that the addition of a partial operator in the model typically means that a scan is added in the algorithm. Moreover, the scans of the algorithm occur sequentially and in a strict order, in order for the various dependencies, which are inherent to the summarize and jump strategy, to be satisfied. For instance, the biomechanical simulation in the tissue scale follows the cellular simulation, since the implementation of the cytokinetic diagram changes the cell populations in the various cell phases and in each geometrical cell, in general, which in turn dictates the expansion or the shrinkage of the tumor.

In that way, the application of the composite operator $f$, as a conceptual cluster of partial operators, which are applied sequentially and in an order that is strict (non commutative), corresponds to the application of a full set of scans for one time step of the algorithm.

### 2.3.4 The simulation input and output parameters, the natural processes that they describe and their role in the Oncosimulator algorithm

As mentioned various times before (**Figure 2.6**, **Figure 2.7**, **Figure 2.8** and **Figure 2.9**), the Oncosimulator operates with reference to a set of parameters, which have the purpose of fully defining the components of the simulation, as well as quantifying and expressing the outcome of the simulation. In that way, the Oncosimulator parameters can be grouped into two categories, the input parameters and the output parameters, respectively.

The **input parameters** of the Oncosimulator are shown in **Table 2.1**, along with the primary scans of the algorithm they are referenced in and a brief description. Their role is to fully define the various stages of the model, so that the simulation can be practically performed with actual patient data. The set of input parameters that is presented here is common for all the versions of the Oncosimulator, since it contains all the necessary input parameter that are needed for the Oncosimulator to function, but this set can be enriched with more parameters in the various versions of the Oncosimulator, depending on the demands of the model, in order for the simulation to be performed as wanted.

| Symbol | Scan | Definition |
|---|---|---|
| $T_d$ | 1 | doubling time of the tumor |
| $CKR$ | 2 | percentage of fatally hit by therapy cells |
| $T_c$ | 3 | cell cycle duration |
| $T_{G0}$ | 3 | time required for a dormant cell to die through necrosis |
| $T_N$ | 3 | time required to complete necrosis and remove its products from the tumor |
| $T_A$ | 3 | time required to complete apoptosis and remove its products from the tumor |
| $T_{G1}$ | 3 | $G_1$ phase duration |
| $T_S$ | 3 | $S$ phase duration |
| $T_{G2}$ | 3 | $G_2$ phase duration |
| $T_M$ | 3 | $M$ phase duration |
| $R_A$ | 3 | percentage of undifferentiated cells that die by apoptosis per hour (applicable for both STEM and LIMP cells) |
| $R_{ADiff}$ | 3 | percentage of differentiated cells that die by apoptosis per hour |
| $R_{NDiff}$ | 3 | percentage of differentiated cells that die by necrosis per hour |
| $P_{G0toG1}$ | 3 | percentage of undifferentiated cells that leave the $G_0$ phase and re-enter the cell cycle (applicable for both STEM and LIMP cells) |
| $N_{LIMP}$ | 3 | maximum number of mitoses that a LIMP cell can perform before its terminal differentiation |
| $P_{sym}$ | 3 | percentage of STEM cells divided symmetrically |
| $P_{sleep}$ | 3 | percentage of cells entering the $G_0$ phase after mitosis |
| $NBC$ | $4-5$ | default number of biological cells in a geometrical cell |
| $margin\_factor$ | $4-5$ | acceptable percentage of temporary overload or unloading of each geometrical cell |
| $color\_criterion$ | 7 | minimum fraction of dead cells in a geometrical cell to classify it as necrotic |
| $I_{GC}$ | N/A | geometrical cell dimension size |
| $x_{dim}$ | N/A | length of the geometrical mesh |
| $y_{dim}$ | N/A | height of the geometrical mesh |
| $z_{dim}$ | N/A | width of the geometrical mesh |

**Table 2.1:** The simulation input parameters, which fully define the distinct scans of the Oncosimulator algorithm and describe the natural processes and quantities that are represented in silico. The doubling time parameter $Td$ expresses the time that is required for the tumor to double in size and cell population. The $CKR$ parameter expresses the effects of therapy that are simulated in the second scan of the algorithm. Together with the parameters that are referenced in the third scan of the algorithm, which determine the time duration of the phases inside and out of the cell cycle and the rates with which the cells apoptose, transition to and from the resting $G_0$ phase and divide symmetrically, they fully define the transitions of the cytokinetic diagram of **Figure 2.10** and complete the simulation in the cellular level. The $NBC$ parameter defines the default number of biological cells inside each geometrical cell. Together with the $margin\_factor$ parameter, they determine the minimum and the maximum number of biological cells inside each geometrical cell, according to **Equations 2.6, 2.7**, which are referenced in the fourth and the fifth scan of the algorithm. The $I_{GC}$ parameter defines the size of the geometrical cells, while the $x_{dim}$, $y_{dim}$ and $z_{dim}$ parameters define the size of the 3D mesh along each axis.

The first parameter, the doubling time $\boldsymbol{T_d}$ of the tumor, i.e. the time that is needed for the tumor to double in size, spatially and with concern to its cell population, is an input parameter that highly depends on the cancer type that the model simulates. It is used in the first scan of the algorithm, where the tumor is initialized in the 3D discretization mesh using the hypermatrix notation, in order to check if the set of input parameters that are given for the simulation is valid, i.e. is referencing a realizable tumor, with the appropriate doubling time that is given for the specific cancer type to be simulated. More specifically, the doubling time is correlated with five input parameters that are involved in the cellular simulation via the third scan, $T_{G0}$, $R_A$, $P_{G0toG1}$, $P_{sym}$ and $P_{sleep}$, according to **Equation 2.5**.

$$\begin{aligned} e^{(\alpha+R_A)\cdot T_C} &= (1+P_{sym})\cdot(1-P_{sleep}+P_{sleep}\cdot\frac{P_{G0toG1}/T_{G0}}{R_A+1/T_{G0}+\alpha}) \\ T_d\cdot\alpha &= ln(2) \end{aligned} \tag{2.5}$$

where $\alpha$ is the growth rate of the tumor.

**Equation 2.5** is the summary of the conditions for the balanced exponential growth of the tumor with growth rate $\alpha$ (or equivalently doubling time $T_d$). By setting the condition that $\alpha \geq 0$, the resulting condition is **Equation 2.1**, which is the condition for the free growth of the tumor, i.e. for a tumor that does not naturally shrink with time. In the first scan of the algorithm, which is responsible for the initialization of the tumor, the free growth condition must be met, but more strictly, if a specific type of cancer is simulated and its doubling time is known, the condition for balanced exponential growth (**Equation 2.5**), with the specific growth rate $\alpha$ or doubling time $T_d$, is used to check the validity of the input.

The $\boldsymbol{CKR}$ input parameter, i.e. the cell kill ratio, defines the percentage of cells that are hit by therapy. It is used in the second scan, which implements the effects of therapy by hitting the appropriate percentage of cells, according to the $CKR$ parameter, and adjusting the populations of biological cells inside the geometrical cells of the 3D mesh tumor, along with other variables, e.g. the remaining time before each equivalence class of biological cells inside the geometrical cell transitions to the next phase. The $CKR$ parameter depends on the treatment modality that is being simulated and can be further analyzed into component $CKR$ parameters, in cases when mixed therapeutic schemes are simulated, e.g. chemotherapy with radiotherapy.

The following fifteen input parameters are involved in the third scan of the algorithm and, along with the $CKR$ parameter of the second scan, they fully define the transitions of the cytokinetic diagram, as shown in **Figure 2.10**.

The $\boldsymbol{T_c}$ parameter, i.e. the cell cycle duration, is the parameter that defines the time duration that is required for the completion of one full cell cycle, including the completion of all its phases, $G_1$, $S$, $G_2$, $M$. The time duration for each separate phase inside the cell cycle, $\boldsymbol{T_{G1}}$, $\boldsymbol{T_S}$, $\boldsymbol{T_{G2}}$, $\boldsymbol{T_M}$, is usually expressed as a ratio of the cell cycle duration $T_c$, which means that the actual input parameter between these 5 is typically $T_c$. The time duration of the cell cycle, as well as the time duration of each separate phase inside it, are typically measured in hours ($h$).

Every time an adjustment is made to the population of cells inside the equivalence classes of the geometrical cells of the 3D mesh tumor, due to the various transitions of the cytokinetic diagram, the remaining time that has to be completed before the equivalence class transitions to another state of the cytokinetic diagram is also readjusted appropriately using the weighted median, where the cells that enter the current state have the maximum time of the current state, e.g. $T_{G1}$, $T_S$, $T_{G2}$, $T_M$ for the phases inside the cell cycle.

Moreover, whenever the time of the simulation is incremented by one time step, e.g. $1h$, this remaining time is decremented also by one time step. When the remaining time is 0, the equivalence class performs the appropriate transitions, according to the cytokinetic diagram. That way, $T_{G1}$, $T_S$, $T_{G2}$, $T_M$, which are typically expressed as a ratio of $T_c$, are the parameters that define the transitions of the cytokinetic diagram, with concern to the cell cycle phases, $G_1$, $S$, $G_2$, $M$, for the STEM and the LIMP cells alike.

The $\boldsymbol{P_{sym}}$ parameter, i.e. the percentage of cells that are divided symmetrically, is the parameter that defines the portion of biological cells of the $M$ phase that are divided symmetrically, producing two daughter cells that are in the STEM cell category and can keep proliferating for an unlimited amount of times. The remaining percentage, $1 - Psym$, corresponds to asymmetric division, which results in one daughter cell in the STEM cell category and one daughter cell in the LIMP cell category, which will complete the cell cycle for a specific amount of times before its terminal differentiation and ultimate death. This amount of times is specified with the $\boldsymbol{N_{LIMP}}$ parameter, i.e. the maximum number of mitoses that a LIMP cell can perform before its terminal differentiation. Obviously, the $P_{sym}$ parameter refers only to the STEM cell category, while the $N_{LIMP}$ refers only to the LIMP cell category.

The $\boldsymbol{P_{sleep}}$ parameter defines the percentage of cells that enter the resting $G_0$ phase after mitosis, e.g. due to lack of nutrients or oxygen. The remaining percentage, $1 - P_{sleep}$ corresponds to the cells that reenter the cell cycle from the first gap phase, $G_1$. The $P_{sleep}$ parameter refers to STEM and LIMP cells alike.

The $\boldsymbol{R_A}$ parameter, i.e. the percentage of undifferentiated cells that die by apoptosis per hour, defines the portion of cells in the cell phases, inside or out of the cell cycle, that die via spontaneous apoptosis. This rate is typically measured in $h^{-1}$ and applies to both STEM and LIMP cells. The $R_A$ parameter, along with the $CKR$ parameter, defines the transitions towards the apoptosis state in the cytokinetic diagram.

The $P_{sym}$, $N_{LIMP}$, $P_{sleep}$ and $R_A$ parameters define the transitions in the cytokinetic diagram that start from the $M$ cell cycle phase. More specifically, for a STEM cell, first spontaneous apoptosis is applied according to $R_A$. The percentage of cells that does not apoptose performs symmetric or asymmetric division according to $P_{sym}$. The LIMP cells that are created enter the cell cycle in the $G_1$, but as LIMP cells now (right side of the diagram). The STEM cells that are created either reenter the cell cycle in the $G_1$ phase, as STEM cells again (left side of the diagram), or enter the resting $G_0$ phase according to $P_{sleep}$. For a LIMP cell, again apoptosis is applied first, then if $N_{LIMP}$ mitoses have been completed, the cell does not perform mitosis, but is terminally differentiated instead. Else the division is perfromed, but with both resulting cells being LIMP cells now and finally the daughter cells reenter the cell cycle or enter the resting $G_0$ phase according to $P_{sleep}$.

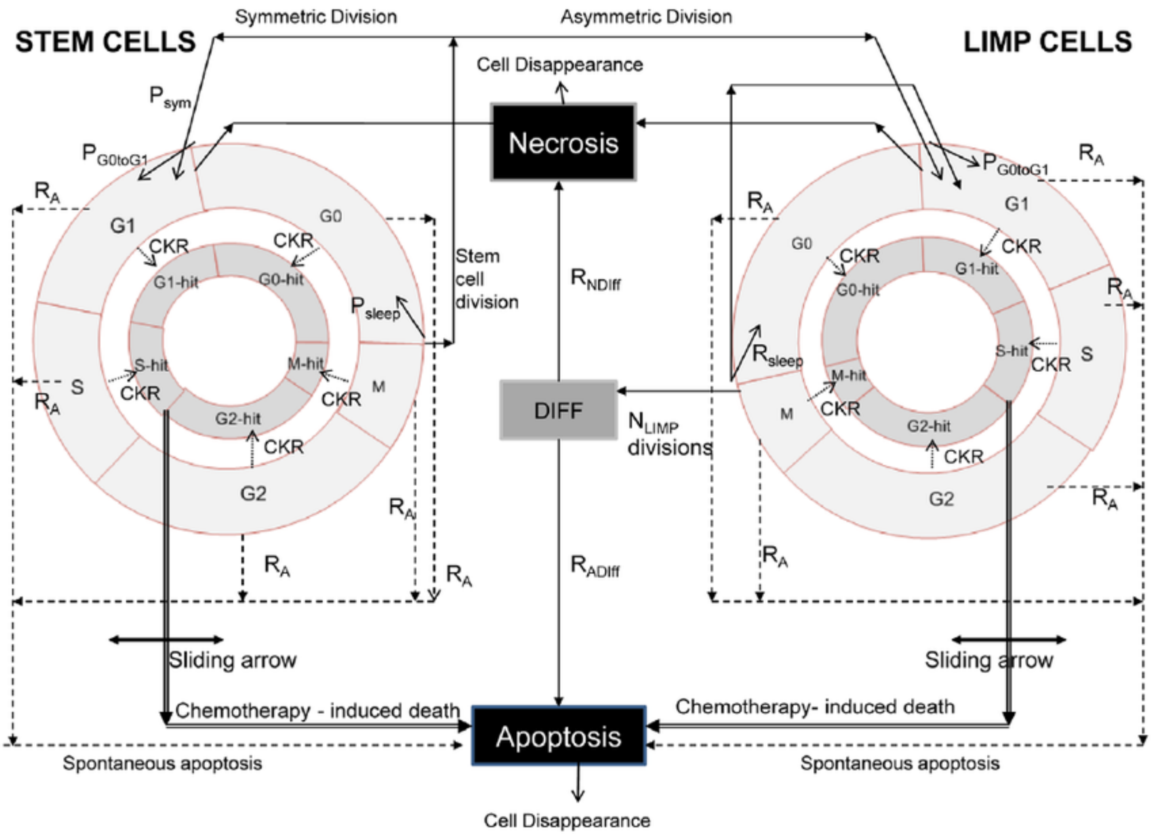


**Figure 2.10:** A demonstration of the manner in which the simulation input parameters that are referenced in the second and third scan of the Oncosimulator algorithm implement the transitions of the cytokinetic diagram [43]. The timing parameters, $T_c$, $T_{G1}$, $T_S$, $T_{G2}$, $T_M$, $T_{G0}$, $T_A$ and $T_N$, describe the maximum time that each cell spends in the corresponding state of the cytokinetic diagram. The $P_{sleep}$ parameter defines the ratio of cells that enter the dormant $G_0$ phase, while the $P_{G0toG1}$ parameter defines the ratio of cells that reenter the cell cycle in the $G_1$ phase. The $P_{sym}$ parameter defines the ratio of cells that perform symmetric division. The $N_{LIMP}$ parameter defines the maximum number of full cell cycle transitions that each LIMP cell performs. The $R_A$ parameter defines the ratio of cells that apoptose spontaneously. The $R_{ADiff}$ and $R_{NDiff}$ parameters define the ratio of differentiated cells that die via apoptosis and necrosis respectively. The $CKR$ parameter defines the ratio of cells that are hit by therapy. The implementation of the cytokinetic diagram with the aid of the aforementioned parameters concludes the simulation in the cellular level, via the completion of the second and third scan of the algorithm for the current time step.

The $\boldsymbol{R_{ADiff}}$ and $\boldsymbol{R_{NDiff}}$ parameters define the percentage of differentiated cells that die by apoptosis and by necrosis, respectively, per hour. As the $R_A$ parameter, they are typically measured in $h^{-1}$. These two parameters define the transitions from differentiation to apoptosis and necrosis, respectively, in the cytokinetic diagram.

The remaining transitions to be defined in the cytokinetic diagram are the ones that start from the resting $G_0$ phase. Except for the spontaneous apoptosis, which is governed by the $R_A$ parameter, there are two other option for the transitions that start from the $G_0$ phase, both for STEM and for LIMP cells. The first option is to reenter the cell cycle in the $G_1$ phase and the second option is to die via necrosis. These two transitions are governed by the $\boldsymbol{P_{G0toG1}}$ parameter and the $\boldsymbol{T_{G0}}$ parameter. More specifically, the $P_{G0toG1}$ parameter defines the percentage of cells in the resting $G_0$ phase that reenter the cell cycle in the $G_1$ phase, while the $T_{G0}$ parameter defines the maximum time that a cell can reside in the resting $G_0$ phase before its death via necrosis. This means that the $G_0$ cells that are not chosen to reenter the cell cycle by the $P_{G0toG1}$ percentage and that have remained in the $G_0$ phase for a time that exceeds the $T_{G0}$ time die via necrosis.

The $\boldsymbol{T_A}$ and $\boldsymbol{T_N}$ parameters define the times that are required for the completion of apoptosis and necrosis, respectively, and the removal of their products from the tumor. As opposed to $T_{G1}$, $T_S$, $T_{G2}$ and $T_M$, the times $T_{G0}$, $T_A$ and $T_N$ are input parameters independent from the $T_c$ parameter.

The $\boldsymbol{NBC}$ parameter, i.e. the number of biological cells, defines the default number of biological cells inside each geometrical cell of the 3D mesh tumor. During the simulation, the actual number of biological cells inside each geometrical cell changes dynamically, but the minimum and maximum number of biological cells inside the geometrical cells are constant. More specifically, they are defined by the $NBC$ parameter and the $\boldsymbol{margin\_factor}$ parameter, i.e. the acceptable percentage of temporary overload or unloading of each geometrical cell, as shown in **Equation 2.6** and **Equation 2.7**. These two parameters are referenced in the biomechanical part of the simulation, in the fourth and the fifth scan, which are responsible for the spatial expansion and shrinkage of the tumor.

$$NBC_{min} = margin\_factor \cdot NBC \tag{2.6}$$

$$NBC_{max} = (1 + margin\_factor) \cdot NBC \tag{2.7}$$

The $\boldsymbol{color\_criterion}$ parameter, i.e. the minimum fraction of dead cells in a geometrical cell to classify it as necrotic, is used in the seventh and final scan of the Oncosimulator algorithm, which is responsible for cleaning the tumor from the geometrical cells that contain too few biological cells by categorizing those geometrical cells as necrotic tissue. This is an important step for the 3D reconstrucion of the tumor and the results of the simulation and their illustration in image. More specifically, the $color\_criterion$ parameter defines the threshold of **Equation 2.8**. The geometrical cells that complete the simulation time steps containing a number of biological cells that is below that threshold are categorized as necrotic tissue in the seventh scan.

$$NBC_{threshold} = color\_criterion \cdot NBC \tag{2.8}$$

The final four parameters of **Table 2.1** do not apply to a specific scan of the Oncosimulator algorithm. They are inherent to the patient specific input of the simulation and characterize the simulation as a whole. More specifically, the $\boldsymbol{I_{GC}}$ parameter refers to the size (in $mm$) of the edge of the geometrical cell. In that way, the $I_{GC}$ parameter defines the resolution of the simulation, since, as mentioned previously (see **Chapter 2.3.3**), increase in the size of the geometrical cell causes decrease in the resolution of the simulation. Finally, the $\boldsymbol{x_{dim}}$, $\boldsymbol{y_{dim}}$ and $\boldsymbol{z_{dim}}$ parameters define the size (in geometrical cells) of the 3D tumor along each axis.

The **output parameters** of the simulation are presented in **Table 2.2**, along with a brief description of the quantity that they express. Their role is to quantify the output of the simulation in a simple and exact manner, so that the results of the simulation are expressed clearly and can consequently be assessed effectively and accurately.

Note that there is an interdependence between the input and output parameters of the Oncosimulator. More specifically, the outcome of the simulation and thus the values for its output parameters depend on the values of the input parameters. In the same way, specific goals for the simulation are expressed in specific values for the output parameters, which in turn affect the values of the input parameters of the simulation. This method is utilized in the clinical adaptation process. All in all, the input and output parameters of the Oncosimulator define a set of interdependent variables and the input parameters should not be superficially considered the free variables of this set. The free variables of the simulation parameter set rather depend on the exploration method that is being performed.

| Symbol | Definition |
|---|---|
| $PROLIF_{initial}$ | initial percentage of proliferative cells |
| $DORMANT_{initial}$ | initial percentage of dormant cells |
| $DIFF_{initial}$ | initial percentage of differentiated cells |
| $DEAD_{initial}$ | initial percentage of dead cells |
| $PROLIF_{final}$ | final percentage of proliferative cells |
| $DORMANT_{final}$ | final percentage of dormant cells |
| $DIFF_{final}$ | final percentage of differentiated cells |
| $DEAD_{final}$ | final percentage of dead cells |
| $-DV$ | relative volume reduction percentage |

**Table 2.2:** The simulation output parameters, which quantify the outcomes of the Oncosimulator algorithm. These outcomes are described by the relative tumor volume reduction percentage $-DV$, as well as by the composition of the initial and the final tumor with respect to the percentage of cell populations, proliferative, dormant, differentiated and dead, that characterize it. The values for these output parameters are dependent on the values for the input parameters. For instance, a high value for the $P_{sleep}$ parameter causes more cells to enter the $G_0$ phase, resulting in more proliferative and less dormant cells. The simulation input and output parameters constitute a set of interdependent variables. The free variables of this set depend on the exploration method that is being performed. For instance, in the clinical adaptation, a goal is set concerning the outcome of the simulation defining the value of some output parameter, e.g. of the $-DV$ parameter, while a number of input parameters, e.g. the $CKR$ parameter, are adjusted in order to satisfy it.

The first four parameters of **Table 2.2** refer to the composition of the initial tumor, i.e. the tumor of the first time step of the simulation that is the product of the initialization step of the simulation algorithm, with respect to the types of cells, proliferative, dormant, differentiated and dead, that it contains and at what percentage each. More specifically, the $\boldsymbol{PROLIF_{initial}}$ parameter expresses the percentage of proliferative cells in the initial tumor, the $\boldsymbol{DORMANT_{initial}}$ parameter expresses the percentage of dormant cells in the initial tumor, the $\boldsymbol{DIFF_{initial}}$ parameter expresses the percentage of differentiated cells in the initial tumor and the $\boldsymbol{DEAD_{initial}}$ parameter expresses the percentage of dead cells in the initial tumor. These percentages depend on:

- the value for the $P_{sleep}$ parameter, whose increase causes more cells to enter the dormant $G_0$ phase, resulting in more dormant and less proliferative cells
- the value for the $P_{sym}$ parameter, whose increase causes a decrease to the production of LIMP cells and consequently less differentiated and more proliferative and dormant cells
- the values for the $T_A$ and $T_N$ parameters, whose increase means the apoptotic and necrotic cells respectively stay longer in the tumor area before they terminally die and exit from it, which in turn causes increase in the differentiated cell population and decrease in the dead cell population
- the value for the $T_{G0}$ parameter, whose increase means that the cells that enter the dormant $G_0$ phase stay there longer, which in turn causes increase in the dormant cell population and decrease in the differentiated cell population
- the values for the $R_{ADiff}$ and $R_{NDiff}$ parameters, whose increase means the differentiated cells become apoptotic and necrotic respectively with a higher rate, which in turn causes decrease in the differentiated cell population and increase in the dead cell population

The following four parameters of **Table 2.2** refer to the composition of the final tumor, i.e. the tumor of the last time step of the simulation after therapy has taken effect and some time steps of free growth have been completed, with respect to the types of cells, proliferative, dormant, differentiated and dead, that it contains and at what percentage each. More specifically, the $\boldsymbol{PROLIF_{final}}$ parameter expresses the percentage of proliferative cells in the final tumor, the $\boldsymbol{DORMANT_{final}}$ parameter expresses the percentage of dormant cells in the final tumor, the $\boldsymbol{DIFF_{final}}$ parameter expresses the percentage of differentiated cells in the final tumor and the $\boldsymbol{DEAD_{final}}$ parameter expresses the percentage of dead cells in the final tumor. These percentages depend on the initial composition of the tumor, as well as on the input parameters that affect the initial composition of the tumor, $P_{sleep}$, $P_{sym}$, $T_A$, $T_N$, $T_{G0}$, $R_{ADiff}$ and $R_{NDiff}$, as they were described in the previous paragraph. Moreover, the final composition of the tumor is significantly affected by the value for the $CKR$ parameter, since greater value for the $CKR$ parameter causes more cells to apoptose and finally disappear, which in turn leads to the decrease of the proliferative and dormant cell population and the increase of the differnetiated and dead cell population.

The final output parameter, $\boldsymbol{-DV}$, expresses the relative tumor volume reduction, i.e. the percentage by which the volume of the final tumor has shrinked compared to that of the initial tumor. The minus sign is used in the $-DV$ notation to indicate that the value for the relative tumor volume reduction percentage is the opposite number of the value for the relative tumor volume variation $DV$ (**Equation 2.9**).

$$\begin{aligned} DV &= \frac{\text{final tumor volume} - \text{initial tumor volume}}{\text{initial tumor volume}} \cdot 100 \\ -DV &= \frac{\text{initial tumor volume} - \text{final tumor volume}}{\text{initial tumor volume}} \cdot 100 \\ DV &= -(-DV) \end{aligned} \tag{2.9}$$

## 2.4 The Nephroblastoma Oncosimulator

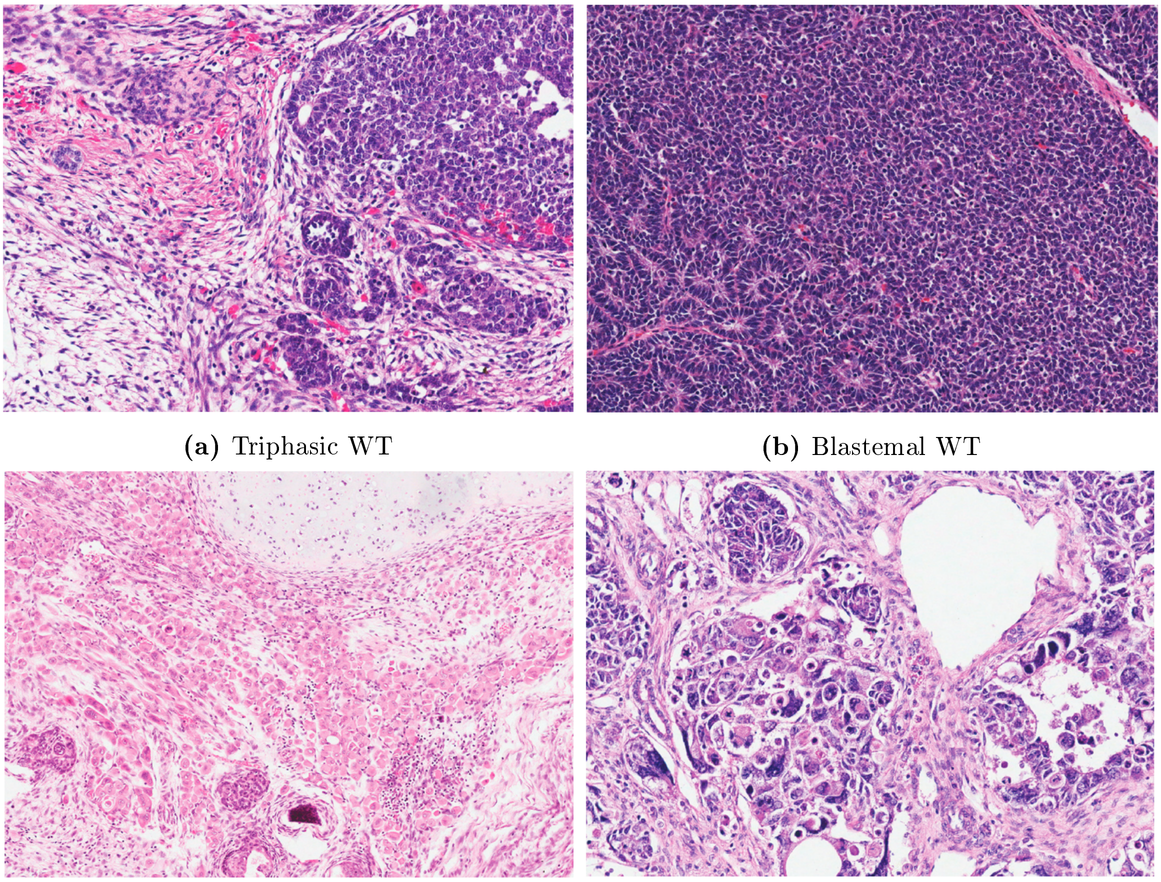
### 2.4.1 A summary of the Wilms' tumor disease, with focus to its histopathology, etiology, prognosis within the standard staging system and epidemiology

Wilms' tumor (WT), also known as nephroblastoma, is a cancer that manifests with the appearance of one or multiple malignant tumors on the kidneys, the organs that are responsible for managing the waste products of metabolism throught the process of receiving blood by the renal arteries, filtering it and returning the clear blood to the circulation via the renal veins and expelling the waste products in the form of urine into the bladder via the ureter. This type of tumor, named after the German surgeon Max Wilms who first described it, occurs more often in childer and rarely in adults, with early symptoms being fever, blood in the urine and abdominal pain. The causes behind the occurence of nephroblastoma are, as with most types of cancer, complex and can be attributed to an interplay of genetic and environmental factors. These causes can be categorized as syndromic, when nephroblastoma co-occurs with a genetic syndrome, or as non-syndromic, when tumor appearance is separate from an existing syndrome. WT is highly treatable with a very good prognosis in most cases. However, the patient has to deal with the short-term and long-term side effects of the administered therapy and the risk of relapse is always present, which leaves great room for progress when it comes to the way that WT is treated, so that the survival rate becomes even higher and the life expectancy and the quality of the patient lives better.

**Pathology/Histopathology** WT usually manifests as one monocentric neoplasm in one of the two kidneys. However, there exists a significant percentage of cases when one or more tumors develop in one or both of the kidneys. When only one kidney is afflicted, either with a monocentric tumor or with multifocal neoplasms, then the disease is characterized as unilateral, while if both the kidneys are afflicted the disease is characterized as bilateral. Most cases of WT develop from nephrogenic rests, which are benign foci of embryonal kidney cells that abnormally persist into postnatal life [44]. Nephrogenic rests have been located in many unilateral tumors (35%) and in significantly more bilater tumors (nearly 100%) [45, 46]. The incidence of nephrogenic rests is associated with mutations in the WT1 gene, which is associated with genetic syndromes that are accompanied by a great risk of multifocal WT occurence. As for the histopathology, WT is a complex embryonal neoplasm arising from metanephric blastema, which typically exhibits triphasic epithelial, blastemal, and stromal differentiation [47]. The proportion and the degree of maturation of the different types of WT tissue, epithelial, blastemal and stromal, vary significantly among the patient population, making the histological appearance of each tumor unique. A single tumor can be monophasic, biphasic ot triphasic, meaning that it can contain one, two or three types of tissue, respectively [47]. The cases of monophasic WT, especially in small specimen, complicate the diagnosis and can lead to a misdiagnosis of a differential diagnosis condition [48]. Blastema represents the least differentiated, and presumed most malignant, component [48] and depending on the evolution of the tumor and its histologic profile, its histology can become anaplastic, in the case of extreme cellular pleomorphism (anaplasia) [47]. Anaplasia is characterized by:

- marked enlargement of nuclei within the stromal, epithelial, or blastemal cell lines (excepting skeletal muscle cells) to at least three times the diameter of adjacent nuclei of the same cell type,
- obvious hyperchromatism of the enlarged nuclei
- and multipolar mitotic figures [47]

Moreover, anaplasia can be categorized as focal, when it is found in foci inside the tumor, or as diffusive, when it also affects the tumor microenvironment [47, 48]. Anaplasia is considered a high risk histology and the tumors that do not develop anaplasia are chategorized as favorable histology tumors.


**(a)** Triphasic WT

**(b)** Blastemal WT

**(c)** Preoperatively treated WT

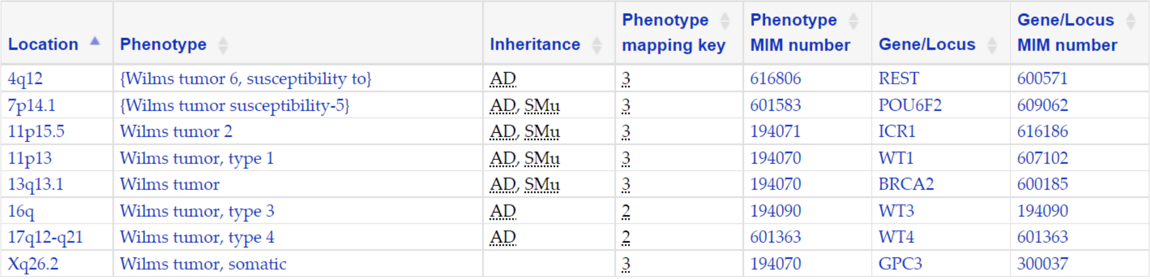
**(d)** Anaplastic histology WT

**Figure 2.11:** A demonstration of different histopathologic scenarios for WT [48].

**Etiology** The incidence of WT can be attributed to a variety of gene mutations that are often associated with the manifestation of further anomalies of the genitourinary system. The primary gene that is linked to the WT phenotype is the WT1 gene, which encodes a zinc finger DNA-binding protein that acts as a transcriptional activator or repressor depending on the cellular or chromosomal context [49]. WT1, which is required for normal formation of the genitourinary system and mesothelial tissues [50], has been mapped to the 11p13 chromosomal location, whose hemizygous deletion has been associated with WT, aniridia, genitourinary abnormalities, and mental retardation (WAGR syndrome) [51]. Mutations and deletion in the WT1 gene have also been found in sporadic non-syndromic WT cases, as well as in blastemal cells of patients with Denys-Drash syndrome [52]. Other gene mutations that are linked to the WT phenotype include:

- mutations in the 11p15 chromosomal location, which contains genes that are expressed by one distinct parent allele and the WT2 gene, whose mutation is associated with Beckwith-Wiedemann syndrome (BWS) and the corresponding occurence of embryonic neoplasms, such as WT [53]
- mutations in the Xq26.2 chromosomal location and the WTX gene, which is mutated somatically in approximately 30% of WT incidences [54]
- mutations in the 17q12-q21 chromosomal location and the FWT-1 gene, which is associated with familial WT
- mutations in the 7p14.1, 13q13.1, 4q12, 16q chromosomal locations

A brief presentation of the locations and the corresponding genes whose mutations are associated with WT occurence is demonstrated in **Figure 2.12**. More information can be found on `https://omim.org/`.



| Location ▲ | Phenotype | Inheritance | Phenotype mapping key | Phenotype MIM number | Gene/Locus | Gene/Locus MIM number |
|---|---|---|---|---|---|---|
| 4q12 | {Wilms tumor 6, susceptibility to} | AD | 3 | 616806 | REST | 600571 |
| 7p14.1 | {Wilms tumor susceptibility-5} | AD, SMu | 3 | 601583 | POU6F2 | 609062 |
| 11p15.5 | Wilms tumor 2 | AD, SMu | 3 | 194071 | ICR1 | 616186 |
| 11p13 | Wilms tumor, type 1 | AD, SMu | 3 | 194070 | WT1 | 607102 |
| 13q13.1 | Wilms tumor | AD, SMu | 3 | 194070 | BRCA2 | 600185 |
| 16q | Wilms tumor, type 3 | AD | 2 | 194090 | WT3 | 194090 |
| 17q12-q21 | Wilms tumor, type 4 | AD | 2 | 601363 | WT4 | 601363 |
| Xq26.2 | Wilms tumor, somatic | | 3 | 194070 | GPC3 | 300037 |

**Figure 2.12:** The chromosomal locations and the corresponding genes that are associated with the WT phenotype [55]. Mutations in the WT1 gene cause sporadic non-syndromic WT cases, as well as syndromic disease in the context of the WAGR and Denys-Drash syndromes. Mutations in the WT2 gene are linked with the syndromic occurence of WT within the Beckwith-Wiedemann syndrome. Mutations in the FWT-1 gene are linked with familial WT.

**Prognosis** The prognosis is fairly good for most of nephroblastoma cases, with the overall survival being approximately 90% [56, 57]. The patient specific prognosis depends greatly on the histology of the neoplasm (favorable or anaplastic), as well as on the stage of the disease, which is defined by the microenvironment of the primary neoplasm in the kidney and the general abdomenopelvic region, the metastasis status of the disease, as well as its status as unilateral or bilateral. Staging, i.e the process of determining the WT stage at the time of diagnosis, is a very important step towards a more approriate and subsequently more effective management that grants the best survival rate with the least possible complications. The WT stages are five and are summarized in **Table 2.3**. The stages are ranked from least to most severe, with the first stage describing the unilateral disease with no metastasis scenario with a tumor that is confined inside the kidney and has not ruptured or afflicted the renal capsule. The second stage is more severe in that it includes affliction of the perirenal soft tissues. Still, the tumor is fully excised in the second stage, which is not the case in the third stage, where remnants of the tumor exist after surgery, either because the tumor ruptured or it has afflicted or penetrated through the peritoneal surface or it extends to vital structures or over the surgical margins and is not resectable. The fourth stage describes a metastatic disease, where cancer has reached lung, liver, bone and/or brain and finally the fifth stage descibes the bilateral disease scenario.

| Stage | Characteristics |
|---|---|
| Stage I | Tumor limited to kidney and completely excised. The surface of the renal capsule is intact. Tumor was not ruptured before or during removal. There is no residual tumor apparent beyond the margins of excision. |
| Stage II | Tumor extends beyond the kidney, but is completely excised. There is regional extension of the tumor, i.e. penetration through the outer surface of the renal capsule into perirenal soft tissues. Vessels outside the kidney substance are infiltrated or contain tumor thrombus. There is no residual tumor apparent at or beyond the margins of excision. |
| Stage III | Residual nonhematogenous tumor confined to abdomen. Any one or more of the following occur: (a) lymph nodes on biopsy are found to be involved in the hilus, the periaortic chains or beyond, (b) there has been peritoneal contamination by tumor such as by biopsy or rupture of the tumor before or during surgery, or by tumor growth that has penetrated through the peritoneal surface, (c) implants are found on the peritoneal surfaces, (d) the tumor extends beyond the surgical margins either microscopically or grossly, (e) the tumor is not completely resectable because of local infiltration into vital structures |
| Stage IV | Hematogenous metastases. Deposits beyond Stage III, i.e. lung, liver, bone and brain. |
| Stage V | Bilateral renal involvement at diagnosis. An attempt should be made to stage each side according to the above criteria on the basis of extent of disease prior to biopsy. |

**Table 2.3:** The staging system for WT [57]. The stage of the disease is defined at the time of diagnosis. The stages are listed from least to most severe. The fifth stage describes the bilateral disease scenario. The fourth stage describes the metastatic disease scenario. The first three stages refer to a unilateral non-metastatic disease. At third stage tumor tissue remains after surgery, while in the first two stages the tumor is fully excised. At first stage only the kidney is afflicted, while at second stage the neoplasm extends to the perirenal soft tissue.

The effect that the histology of the neoplasm and the stage of the disease have on the outcomes of the treatement for WT patients is expressed in **Table 2.4**, which displays the 10 year outcomes of the WT patients treated on the National Wilms Tumor Study (NWTS) trials, with respect to the Relapse Free Survival (RFS) rates and to the Overall Survival (OS) rates. Indeed, as the stage of the disease increases the survival rates decrease. Even more dramatic is the effect of the histology, with the anaplastic histology reducing the survival rates by a factor of more than four compared to the favorable histology corresponding rates, when combined with an increased disease stage.

| Histology | Stage | 10 year RFS% | 10 year OS% |
|---|---|---|---|
| Favorable | Stage I | 91 | 96 |
| Favorable | Stage II | 85 | 93 |
| Favorable | Stage III | 84 | 89 |
| Favorable | Stage IV | 75 | 81 |
| Favorable | Stage V | 65 | 78 |
| Anaplastic | Stage I | 69 | 82 |
| Anaplastic | Stage II-III | 43 | 49 |
| Anaplastic | Stage IV | 18 | 18 |

**Table 2.4:** The 10 year outcomes for the WT patients treated on NWTS-4, along with the corresponding histology and disease stage [57]. Increase in the disease stage causes decrease in the survival rates. The anaplastic histology causes even more dramatic decrease in the rates compared to the corresponding favorable histology ones.
RFS: Relapse Free Survival, OS: Overall Survival

| Syndrome | Risk factor for WT | Median ages of WT occurence |
|---|---|---|
| Beckwith-Wiedemann Syndrome | 4.1% | 24 months |
| Hemihypertrophy | $3 - 4\%$ | 37 months |
| Bohring-Optiz | 6.9% | 24 months |
| Mulibrey | 6.7% | 30 months |
| Perlman | 75% | $< 24$ months |
| Simpson-Golabi Behmel | 8% | Undefined |
| Trisomy 18 | $> 1\%$ | 68 months Most $5 - 9$ years |
| WAGR | 50% | 22 months Most $< 8$ years |
| Denys-Drash | $> 90\%$ | 12 months Most $< 3$ years |
| Frasier | Several cases | Undefined |

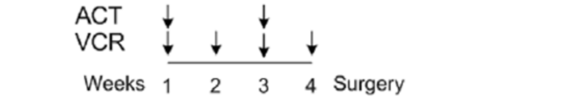
**Table 2.5:** The overgrowth syndromes that are related to the occurence of WT, along with the corresponding risk factors and the median ages of occurence [58]. Many of these syndromes have the same genetic cause as WT, e.g. WT1 for WAGR and Denys-Drash and WT2 for Beckwith-Wiedemann.

**Epidemiology** WT is more prevalent in children. From the 2342 cases that were identified in the U.S. between 1973 and 2005, only 152 were adult with the 2190 rest being paediatric. The female patients are slightly more frequent in the adult cases (60.5% vs 39.5%), while the prevalence does not seem to be affected by the sex in the paediatric cases. The disease is in most cases (92.6%) unilateral, with the right and the left kidney being afflicted with the same frequence [59]. Moreover, as mentioned previously, the cause for the insidence of WT can be syndromic, meaning that the tumor can result from an existing genetic syndrome. A set of overgrowth syndromes that are linked to the occurence of WT are presented in **Table 2.5**, along with the corresponding risk factor for developing WT and the median age of WT occurence. The etiology of many syndromes of **Table 2.5** are associated with the gene mutations that have been found to cause WT (e.g. WT1 for WAGR and Denys-Drash and WT2 for Beckwith-Wiedemann).

### 2.4.2 The treatment plan specific parameters and the reference values for the complete set of simulation input parameters

The Nephroblastoma Oncosimulator is a particular version of the generic Oncosimulator schema, as it was presented in the dedicated chapter, which is enhanced in order to simulate the Wilms' tumor disease in silico. More specifically, the Nephroblastoma Oncosimulator represents the solid kidney tumor in silico as a 3D matrix of discrete geometrical cells. Each geometrical cell is composed of a multitude of biological cells that perform the transitions of the cytokinetic diagram (see **Figure 2.8**) organized in equivalence classes, according to the hypermatrix notation (see **Figure 2.9**). The evolution of the solid tumor in time is simulated according to the "summarize and jump" algorithm of **Figure 2.7**, which is fully defined by the set of input parameters of **Table 2.1**.

In order for the Nephroblastoma Oncosimulator model and algorithm to be fully defined, they need to be enriched with the parameters that describe the disease specific treatment plan. The treatment plan that is simulated in the context of the Nephroblastoma Oncosimulator is the standard 4-week chemotherapy treatment plan of the SIOP protocol, as it is summarized in **Figure 2.13**. The regimen consists of the administration of two chemotherapeutic drugs, Actinomycin (ACT) and Vincristine (VCR), at 4 time points that are evenly distributed over the course of four weeks. More specifically, at the time points of the first and third week, both drugs are administered, while at the time points of the second and fourth week, only Vincristine is administered. The dose of each drug is predetermined, but they can change depending on patient's weight and therapy response. This treatment plan is applied preoperatively and is followed by the surgical removal of the remaining shrinked tumor.



ACT: Actinomycin-D. 45μg/kg i.v. bolus injection (max 2000μg)
VCR: Vincristine. 1.5mg/m$^2$ i.v. bolus injection (max 2.0mg)

If body weight < 12kg: dose reduction to 2/3 for each drug
Major intolerance: doses on the next course should be reduced to 2/3

**Figure 2.13:** The mixed chemotherapeutic scheme of the standard SIOP protocol, that is applied for the preoperative treatment of Wilms' tumor [43]. The two drugs, Actinomycin (ACT) and Vincristine (VCR), are applied at four time points that are evenly distributed along a period of four weeks. At the first and third week both drugs are administered, while at the second and fourth week only Vincristine is administered. The doses may be changed, according to the patient's weight and therapy response.

Since the simulated therapeutic scheme encompasses the administration of two distinct drugs, the $CKR$ parameter that expresses the effects of therapy (see **Table**

**2.1**) can be analyzed into two components, one for each drug. The relationship between the cell kill ratio for the Vincristine drug ($CKR\_VCR$), the cell kill ratio for the Actinomycin drug ($CKR\_ACT$) and the total cell kill ratio ($CKR\_TOTAL$) is described in **Equation 2.10** and is indicative of the percentage with which each drug participates in the total effects of the therapy.

$$\begin{aligned} CKR\_TOTAL &= CKR\_ACT + CKR\_VCR \\ CKR\_ACT &= \frac{3}{5} \cdot CKR\_TOTAL \\ CKR\_VCR &= \frac{2}{5} \cdot CKR\_TOTAL \end{aligned} \tag{2.10}$$

The set of simulation input parameters for the Nephroblastoma Oncosimulator is completed with the parameters that define the key time points of the simulation, as they are summarized in **Table 2.6**. These parameters define the time points of drug administration in the simulation, with compliance to the standard SIOP protocol treatment plan of **Figure 2.13**, as well as the time point at which the simulation is completed.

| Symbol | Definition |
|---|---|
| $VCR\_ADMIN\_A$ | The first time point of Vincristine administration within the standard SIOP chemotherapeutic scheme |
| $VCR\_ADMIN\_B$ | The second time point of Vincristine administration within the standard SIOP chemotherapeutic scheme |
| $VCR\_ADMIN\_C$ | The third time point of Vincristine administration within the standard SIOP chemotherapeutic scheme |
| $VCR\_ADMIN\_D$ | The fourth time point of Vincristine administration within the standard SIOP chemotherapeutic scheme |
| $ACT\_ADMIN\_A$ | The first time point of Actinomycin administration within the standard SIOP chemotherapeutic scheme |
| $ACT\_ADMIN\_B$ | The second time point of Actinomycin administration within the standard SIOP chemotherapeutic scheme |
| $DT\_post\_treat$ | The time duration between the last drug administration time point and the simulation completion time point |

**Table 2.6:** The parameters that define the key time points of the simulation and that, along with the parameters of **Table 2.1**, complete the set of input parameter for the Nephroblastoma Oncosimulator. The time points of drug administration, as well as the duration between the last drug administration time point and the simulation completion time point are defined.

The reference values for the set of simulation input parameters of the Nephroblastoma Oncosimulator are displayed in **Table 2.7**. These reference values are assumed by the relevant bibliography, while the values for the parameters of **Table 2.6** are assigned the values that strictly implement the treatment plan of **Figure 2.13**. In the case that the treatment schedule slightly differs, e.g. the patient missed an appointment and the corresponding drug administration was shifted a few days, the values for the simulation time point parameters are adjusted accordingly.

| Parameter | Refrence Value | Bibliography |
|---|---|---|
| $T_d$ | 29 | [43] |
| $T_c$ | 23 | [60] |
| $T_{G0}$ | 96 | [61] |
| $T_N$ | 20 | [62] [63] [64] |
| $T_A$ | 6 | [65] [66] |
| $R_A$ | 0.001 | [65] [66] |
| $R_{ADiff}$ | 0.003 | [65] [66] |
| $R_{NDiff}$ | 0.001 | [62] [63] [64] |
| $P_{G0toG1}$ | 0.01 | |
| $N_{LIMP}$ | 3 | |
| $P_{sym}$ | 0.45 | |
| $P_{sleep}$ | 0.28 | |
| $CKR\_VCR$ | 0.3 | [67] [68] |
| $CKR\_ACT$ | 0.2 | [69] [70] |
| $CKR\_TOTAL$ | 0.5 | |
| $VCR\_ADMIN\_A$ | 4 | |
| $VCR\_ADMIN\_B$ | 11 | |
| $VCR\_ADMIN\_C$ | 18 | |
| $VCR\_ADMIN\_D$ | 25 | |
| $ACT\_ADMIN\_A$ | 4 | |
| $ACT\_ADMIN\_B$ | 18 | |
| $DT\_post\_treat$ | 3 | |

**Table 2.7:** The reference values for the set of simulation input parameters of the Nephroblastoma Oncosimulator.

### 2.4.3 A brief presentation of the dataset patients, with concern to the plan with which they were treated, their medical imaging data, their histologic profile and the risk group that they represent

The patient data that were utilized in the present work correspond to actual patient cases and were provided by the University Hospital of the Saarland. More specifically, the patient dataset consists of three distinct patients, each one representing a unique patient group, with concern to the histological subtype and the overall prognosis and risks. Each dataset patient is matched to a unique $ID$ number, so that the corresponding data can be referenced and analyzed with compliance to the General Data Protection Regulation (GDPR). Moreover, each dataset patient is given a unique pseudonym, for the purposes of enhancing the simplicity of the simulation executions and of the evaluation of the subsequent simulation outcomes.

After being diagnosed with WT, the three dataset patients underwent the preoperative treatment of the SIOP protocol, according to the standard one-month Vincristine and Actinomycin regimen. After the chemotherapy treatment, the tumor was surgically removed and its histology was analyzed. Moreover, medical imaging data in the form of Magnetic Resonance Tomography (MRT) were acquired for each patient, both before and after the preoperative treatment, so that the response to the chemotherapy and the subsequent overall risk could be evaluated.

The histopathology data for each dataset patient, as they were specified postoperatively via the examination of the neoplasm tissue that was acquired during the tumor removal surgery, are displayed in **Table 2.8**.

Patient_15 corresponds to a monophasic tumor case that is comprised solely by epithelium tissue. This is considered to be a favorable histology, a fact that is mirrored in the percentage of regressive changes in the tumor, which was evaluated macroscopically at the very high value of 98%.

On the other hand, Patient_16 corresponds to a triphasic tumor case that is mainly blastemal. This is considered to be an unfavorable histology that leads to a low percentage of macroscopic regressive changes in the tumor, equal to 5%.

Finally, Patient_17 corresponds to a medium risk histology tumor, that is biphasic comprising primarily by stromal tissue (80%) and at a lower percentage by blastemal tissue (20%). The macroscopic regressive changes in the tumor are evaluated at 70%, which is a satisfying percentage, but not as impressive as the one for Patient_15.

According to the histology data presented in **Table 2.8**, each patient of the dataset represents a distinct patient group with respect to the histopathology of the neoplasm and the subsequent risks. More specifically, Patient_15 represents the low risk favorable histology patient cases, Patient_16 represents the high risk unfavorable histology patient cases and Patient_17 represents the medium risk patient cases that have a mixed histology.

| **Pseudonym** | Patient_15 | Patient_16 | Patient_17 |
|---|---|---|---|
| **ID** | 12180 | 12358 | 12794 |
| **Necrosis / regressive changes (macroscopic)** | 98% | 5% | 70% |
| **Necrosis / regressive changes (histological)** | - | - | - |
| **vitale blastema** | 0% | 70% | 20% |
| **vitale epithelium** | 100% | 5% | 0% |
| **vitale stroma** | 0% | 25% | 80% |

**Table 2.8:** The histopathology data for each dataset patient. Patient_15 corresponds to a favorable monophasic histology tumor that shows high regressive changes. Patient_16 corresponds to an unfavorable triphasic histology tumor that shows low regressive changes. Patient_17 corresponds to a biphasic histology tumor that shows satisfying regressive changes. Each of the three patients represents a distinct patient group, Patient_15 the low risk patients, Patient_16 the high risk patients and Patient_17 the medium risk patients.

A brief presentation of the treatment plan that was fulfilled for each patient of the dataset, containing the dates of the preoperative treatment plan, the MRT acquisition dates and the date of the tumor operation, is displayed in **Table 2.9**.

Both Patient_16 and Patient_17 were treated with the standard preoperative regimen of the SIOP protocol, which requires the administration of Vincristine and Actinomycin in four time points during the period of one month. More specifically, at the first and third time point both Actinomycin and Vincristine are administered, while at the second and the fourth time point only Vincristine is administered. The four time points are separated by seven days (roughly) from one another, placing the first time point in the first week, the second time point in the second week and so on, causing the preoperative treatment to last one month (roughly).

Unlike Patient_16 and Patient_17, Patient_15 represents the metastatic disease scenario. As a result, the applied preoperative regimen for Patient_15 differs from the one applied for Patient_16 and Patient_17, in order to address the metastasis. More specifically, Patient_15 is treated with a 6-week variation of the standard 4-week preoperative regimen that includes the drug Doxorubicin. At the first and fifth week all three drugs are administered, at the second, fourth and sixth week only Vincristine is administered and at the third week both Actinomycin and Vincristine are administered.

| **Pseudonym** | Patient_15 | Patient_16 | Patient_17 |
|---|---|---|---|
| **ID** | 12180 | 12358 | 12794 |
| **Image Pre-Chemo** | 07.09.2011 | 04.02.2013 | 22.08.2016 |
| **Start of Chemo** | 13.09.2011 | 06.02.2013 | 01.09.2016 |
| **Image Post Chemo** | 26.10.2011 | 08.03.2013 | 23.09.2016 |
| **Operation** | 02.11.2011 | 19.03.2013 | 27.09.2016 |
| **Treatment Plan** | 13.9.11:AVD,<br>20.9.11:V,<br>29.9.11:AV,<br>6.10.11:V,<br>13.10.11:AVD,<br>20.10.11:V | 6.2.13:AV,<br>13.2.13:V,<br>22.2.13:AV,<br>1.3.13:V | 1.9.16:AV,<br>8.9.16:V,<br>15.9.16:AV,<br>22.9.16:V |

**Table 2.9:** The treatment plan for each dataset patient, along with the dates of the MRT acquisition and the operation date. Patient_16 and Patient_17 are treated with Actinomycin and Vincristine within the standard 4-week regimen of the SIOP protocol. Patient_15 represents the metastatic disease scenario and, as a result, is treated with Actinomycin, Vincristine and Doxorubicin within the 6-week regimen of the SIOP protocol that is specific to metastatic disease cases. The simulation of the different preoperative treatment regimens is performed by setting the appropriate values to the Nephroblastoma Oncosimulator input parameters that define the treatment plan.

The preoperative treatment regimens differ between patients, even within the SIOP protocol, in order to properly address the various disease scenarios and unique patient needs. For instance, here the regimen for Patient_15 differs from the regimen for Patient_16 and Patient_17, because Patient_15 corresponds to the metastatic disease scenario. Even within the same regimen, the regimen realization may differ between patients, in order to accomodate the patient specific needs, e.g. the patient response to the administered drug (side effects etc.). For instance, here the third time point of the preoperative treatment regimen for Patient_16 was shifted two days after the one described in the protocol (22.2.13 instead of 20.2.13). The patient specific regimen realization is simulated in the Nephroblastoma Oncosimulator by setting the appropriate values for the input parameters that define the treatment plan, which enhances the patient specific nature of the simulation (see **Table 2.10**).

| **Pseudonym** | Patient_15 | Patient_16 | Patient_17 |
|---|---|---|---|
| **ID** | 12180 | 12358 | 12794 |
| **VCR-ADMIN-A** | 4 | 4 | 4 |
| **VCR-ADMIN-B** | 12 | 11 | 11 |
| **VCR-ADMIN-C** | 19 | 20 | 18 |
| **VCR-ADMIN-D** | 26 | 27 | 25 |
| **ACT-ADMIN-A** | 4 | 4 | 4 |
| **ACT-ADMIN-B** | 19 | 20 | 18 |
| **DT-post-treat** | 6 | 7 | 1 |

**Table 2.10:** The values for the input parameters of the Nephroblastoma Oncosimulator that define the treatment plan for each dataset patient. The first dose of Actinomycin is administered at the fourth day of the simulation, allowing the free growth of the tumor for the first three days of the simulation. The dates of the drug admnistation correspond to the actual treatment plan, as it is described in **Table 2.9**. The treatment plan for Patient_15 is simulated starting from the drug administration of 29.9.11 and the Doxorubicin drug is not taken into consideration.

The response to the preoperative treatment that was applied for each dataset patient is summarized in **Table 2.11**. The tumor reduces in size as a result of the preoperatively administered chemotherapy for all three patients and, in that respect, all three patients are considered good responders to the administered chemotherapy regimen.

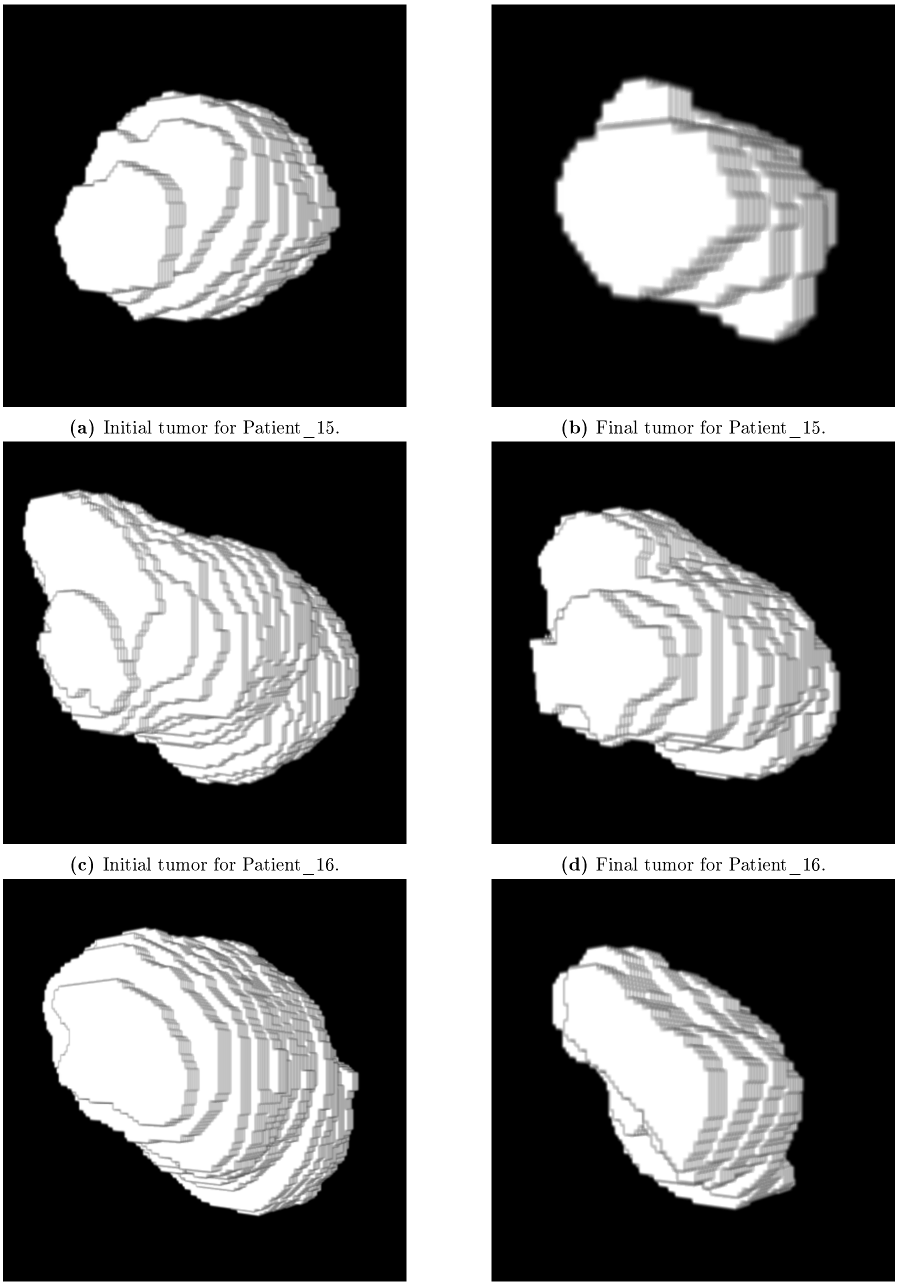
However, their unfavorable tumor histology limits the tumor shrinkage for Patient_16 (tumor volume reduced by $\frac{(285ml-126ml)\cdot 100\%}{285ml} \approx 55.7895\%$) compared to the tumor volume reduction for Patient_15 (tumor volume reduced by $\frac{(144ml-11ml)\cdot 100\%}{144ml} \approx 92.3612\%$). The tumor volume reduction percentage for Patient_17 (tumor volume reduced by $\frac{(536ml-90ml)\cdot 100\%}{536ml} \approx 83.2090\%$) lies between the tumor volume reduction percentages for Patient_16 and Patient_15.

In any case, the tumor is reduced to more than half its initial size for all three dataset patients, which classifies them as good responders to the preoperative chemotherapy regimen. Patient_16 differs from Patient_15 and Patient_17 in that they have an unfavorable tumor histology, which classifies them as high risk, despite their good response to the preoperative therapy.

| **Pseudonym** | Patient_15 | Patient_16 | Patient_17 |
|---|---|---|---|
| **ID** | 12180 | 12358 | 12794 |
| **Good / Bad Responder** | $144ml \rightarrow 11ml$: good response, complete remission of lung metastasis, (regressive subtype) | $285ml \rightarrow 126ml$: good responder, but high risk tumor (blastemal subtype) | $536ml \rightarrow 90ml$: good response (regressive subtype) |
| **Image Pre-Chemo (Database MRT)** | $143987.13mm^3$ | $306985.833mm^3$ | $536733.98mm^3$ |
| **Image Pre-Chemo (Segmentation MRT)** | $130556.25mm^3$ | $247489.53mm^3$ | $554914.55mm^3$ |
| **Image Post-Chemo (Database MRT)** | $10616.9mm^3$ | $126287.764mm^3$ | $89487.915mm^3$ |
| **Image Post-Chemo (Segmentation MRT)** | $10451.07mm^3$ | $108919.51mm^3$ | $70812.7mm^3$ |

**Table 2.11:** The response of the dataset patients to the applied preoperative therapy. The preoperatively applied chemotherapy regimen reduces the tumor volume for all three patients of the dataset. The tumor volume reduction percentage is greater for Patient_15, lower for Patient_17 and minimum for Patient_16, a response that is in accordance with the tumor histology profile for each patient. In any case, the tumor shrinks to more than half its initial size for all three patients, which classifies them as good responders to the preoperative treatment. Patient_16 differs from the other two in that they have an unfavorable tumor histology that classifies them also as high risk.

The 3D images of the initial tumor (Image Pre-Chemo) and the final tumor (Image Post-Chemo) for each dataset patient, as they were captured in MRT at the corresponding dates of **Table 2.9**, are displayed in **Figure 2.14**. The 3D tumors of **Figure 2.14** were constructed using the ImageJ software.



(a) Initial tumor for Patient_15.

(b) Final tumor for Patient_15.

(c) Initial tumor for Patient_16.

(d) Final tumor for Patient_16.

(e) Initial tumor for Patient_17.

(f) Final tumor for Patient_17.

**Figure 2.14:** The visualized with ImageJ 3D reconstruction images of the tumors before and after the preoperative treatment application for each dataset patient.

# Methodology

## 3.1 Workflow Outline

The purpose of the current chapter is to demonstrate the workflow that was implemented in the context of the present work. This workflow is briefly outlined in the form of a series of well-defined discrete steps in the diagram of **Figure 3.1**.

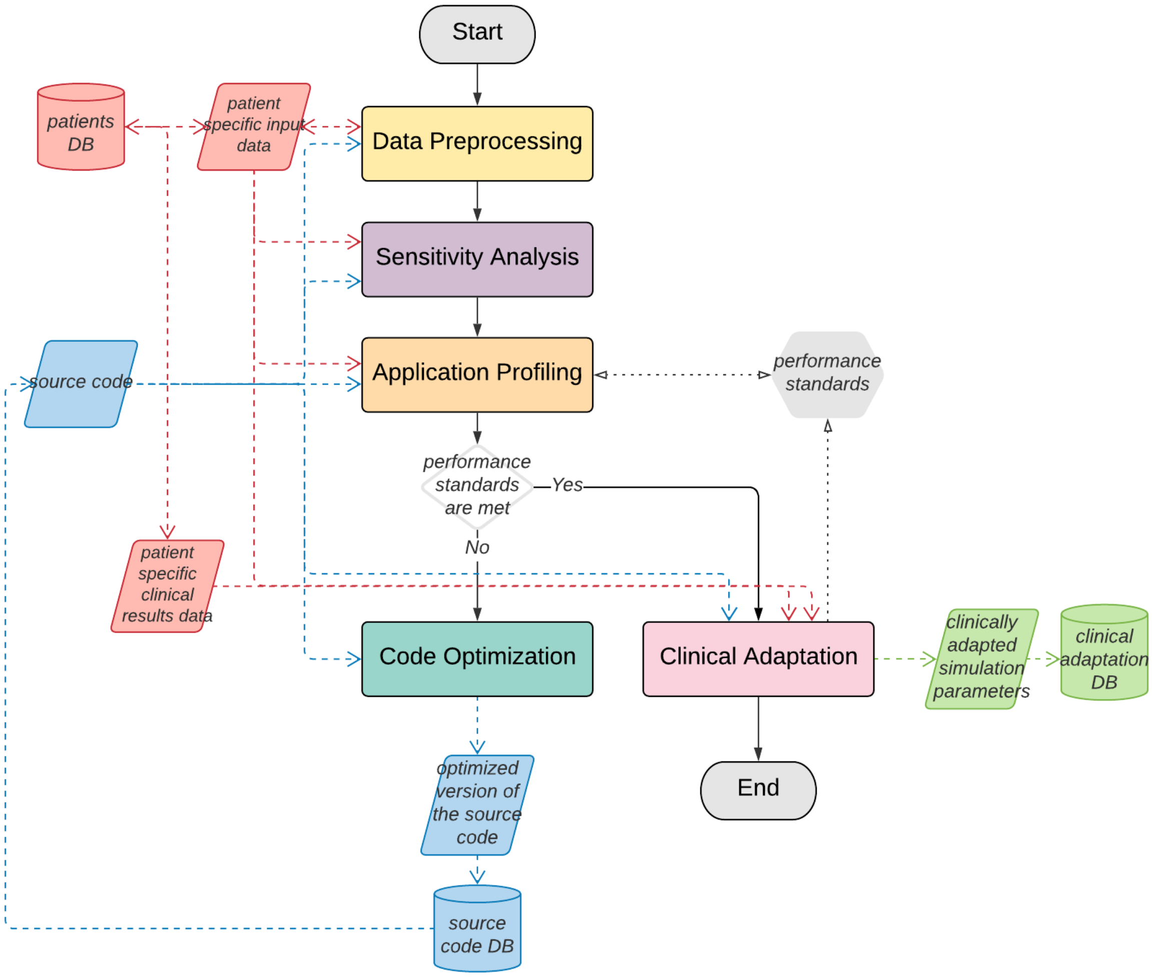


**Figure 3.1:** The diagram of the outline of the workflow that was implemented in the context of the present work. The final desired result of the workflow is the clinical adaptation of the patient parameters. The clinical adaptation step defines the performance standards that the Oncosimulator application should meet. The versions of the source code are continuously optimized in the loop that profiles the performance of the current version and that adjusts it into a new optimized version. Each source code version that is produced also passes through the sensitivity analysis step, in order for its functionality to be verified. At first, the patient data are preprocessed and transformed into a shape that is appropriate to be given as input to the Oncosimulator. Then, the optimized versions of the source code are produced, verified and profiled in the loop of the sensitivity analysis, application profiling and code optimization steps, until the performance standards are met. When the loop is broken, the final clinical adaptation step is executed using the optimal source code version.

The continuous arrows in **Figure 3.1** indicate the sequence of the steps of the workflow, while the dashed arrows indicate the flow of data. As shown in **Figure 3.1**, the workflow utilizes two types of stored data, the input data of the simulation (red DB), which reference the patient specific medical data, and the source code of the simulation (blue DB). The source code is expressed in various versions, each one referencing a specific adjustment and improvement in performance that has been achieved as an intermediate result of the workflow. The final result of the workflow is the clinical adaptation, which is stored in the form of the adjusted simulation parameters for each patient (green DB). It should be noted that no Database Management System (DBMS) has been used and that here the database (DB) diagram symbol is used abusively to suggest that the referenced data are stored data.

**Data Preprocessing** The first step of the workflow is the preprocessing of the patient data. The patient specific medical data that are provided are often given in a complex form that cannot be immediately and efficiently used by the Oncosimulator. This step of the workflow is responsible for the filtering, enrichment, modification and overall transformation of the patient data, so that it can be given as input to the simulation, with the goal of an efficient execution. It is important that this step precedes the rest, so that the performance enhancement needs and prospects of the simulator are evaluated with the optimal performance data-wise ensured. With focus on the performance enhancement, an error window with respect to the results that the simulation with the original input patient data produces may be allowed. That is why the source code is provided in the inputs of the data preprocessing step, in order to compare the simulation results of the modified input data with the original input data and control the consequent error. Also, note that the data preprocessing step both inputs the original patient data from the patient data DB and outputs the modified input data to the patient data DB, thus the bidirectional arrows in **Figure 3.1**. The data preprocessing step is performed only once at the start of the workflow.

**Sensitivity Analysis** The data preprocessing step is followed by the sensitivity analysis step, i.e. the step that examines the sensitivity of the output of the simulation with regard to the variation of the values of the input parameters. As a parameter exploration method, sensitivity analysis has great value with regard to the evaluation of a parametric application, such as the Oncosimulator. In the context of the present work, the sensitivity analysis step is utilized to ensure that the modified patient input data (produced by the data preprocessing step) and the various versions of the source code of the Oncosimulator (produced by the code optimization step) work together as expected. That is why it is crucial that this step follows the data preprocessing step and is repeated for every new version of the source code that is produced by the code optimization step. If the behavior of the simulation for the exploration of the values for the input parameters stays true to that of the original verified version, then the data and code modifications are safe. However, if the parametric behavior deviates from the expected one, then some error that needs to be addressed has occured in the latest modifications.

**Application Profiling** The next step of the workflow is responsible for the profiling of the application, with concern to the performance of the application as well as to the patient dataset that acts as input for the application. The evaluation of the application's performance is crucial before the optimization goals can be revisited and the corresponding optimization techniques considered and explored. More specifically, the process of the performance profiling of the application can determine if the performance standards that have been set are met and if not, if they are achievable or if they should be redefined in order to pursue more plausible goals (bidirectional arrow in **Figure 3.1**). Once the performance goals have been defined accurately, the performance profiling can indicate the performance-wise poor parts of the application, which can then be optimized through various techniques devised and implemented in the next step of the workflow. To that end, it is important that the performance profiling is based on various executions with different inputs, so that it is more fit to represent a diverse set of execution scenarios and so that it is more reliable in general. In this context, it's also useful to profile the patient dataset that is used as input for the application and to observe the correlation of the patient input attributes with the application's execution behavior and its performance attributes, if any correlation exists. The same can be argued for the application's input parameters, which have been adequately profiled in the sensitivity analysis step. However, if a more detailed exploration of the input parameter space is required, it can be incorporated in the profiling step.

**Code Optimization** If the performance standards are not met, then the source code needs to be adjusted in order to better accommodate them. With the information provided via the application profiling that preceded, an optimization plan can be devised with the goal of acomplishing those standards. The role of this step is to plan the code adjustments, as well as to implement them. More specifically, the code optimization step accepts as input an inadequate version of the source code, applies a series of optimizations on it and outputs an optimized version, which is then stored as an intermediate result of the workflow in the source code DB. At a later time, this version is taken from the source code DB and given as input to the sensitivity analysis step where it can be verified with concern to its functionality and to the application profiling step where it can be evaluated with concern to its performance. This loop in the workflow ensures that the original source code is continuously adjusted into discrete versions that funtion appropriately and with each version being the optimization product of the previous version.

**Clinical Adaptation** Once the performance goals have been achieved, the loop that produces optimized source code versions can be broken and the optimal version can be utilized in the final step of the workflow, the clinical adaptation step, which produces the final desired result, the clinical adaptation data per patient. The Oncosimulator simulates the evolution of a tumor given the patient specific medical data and the values for the input parameters (**Table 2.1**). However, the values for the input parameters cannot be accurately derived from the patient specific medical data and given the deviation that characterizes them, which underlines the biological diversity of the patient population, a set of simulations for each patient needs to be performed and the outputs coevaluated using statistical and/or artificial intelligence techniques, in order to find the appropriate set of parameter values for each patient. In order for this process to be performed, the actual clinical results of the patient's tumor evolution are necessary. By performing simulations for a set of virtual patients with each one corresponding to a specific actual patient and to a discrete set of values for the input parameters of the simulation and by combining the outputs of the simulations that produce results that match the actual evolution of the patient image within a minimal supported error window, a window of acceptable values can be derived for each input parameter and for the specific patient. Since the clinical adaptation step is the final desired step of the workflow that uses the intermediate results of the preceded workflow loop, i.e. the optimized versions of the source code, it is also the step that is crucial to the definition of the performance standards that a single simulation, i.e. a single execution of the Oncosimulator, should meet.

In the following sections of this chapter, each discrete step of the workflow of **Figure 3.1** is presented in further detail, along with the techniques and tools that were utilized for their implementation.

## 3.2 Data Preprocessing

### 3.2.1 The input files of the Nephroblastoma Oncosimulator, which define the properties of the 3D matrix that represents the solid tumor in silico

In order for the tumor of a particular patient to be simulated, the patient specific medical imaging data need to be provided. These data are given as direct input to the Nephroblastoma Oncosimulator executable in the form of two files. The role of these files is to describe the morphology of the initial tumor, i.e. of the tumor at the start of the simulation, with respect to its physical size and shape.

The primary input file is a binary file with .raw extension. It contains the raw information about the positions of the 3D discretization mesh that correspond to tumor tissue. More specifically, each byte of the .raw file is matched to a geometrical cell, i.e. an element of the 3D matrix. If the byte is equal to 0xFF (white color), then the corresponding geometrical cell is considered to be part of the tumor tissue. On the other hand, if the byte is equal to 0x00 (black color), then the corresponding geometrical cell is considered to be part of the tumor microenvironment normal tissue. The size of the .raw file depends on the size of the 3D discretization mesh, since it contains one byte per matrix item.

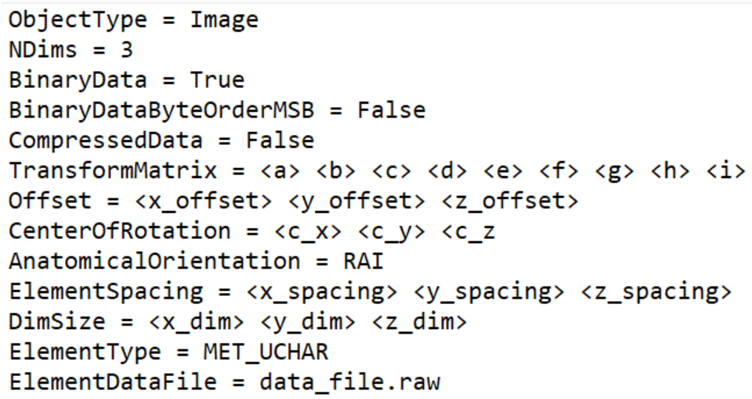
The .raw file is accompanied by a MetaImage MetaHeader file with the characteristic .mhd extension. This file is responsible for providing the header information for the .raw image, such as the size of each dimension of the 3D discretization mesh and the physical size of each geometrical cell. This information is imperative for the interpretation of the .raw file, the accurate description of the initial tumor morphology and its transformation in a 3D matrix for the purposes of the simulation. A standard format for the .mhd file is presented in **Figure 3.2**.

The last row of the .mhd file, with **ElementDataFile** as tag, points out the corresponding .raw file which it acts as header for. The .mhd and .raw files have to be stored under the same directory for the correlation between them to be succesful.

The first two rows, with **ObjectType** and **NDims** as tags, expresse the fact that the .mhd file acts as a header for a 3D image. The third row, with **BinaryData** as key, characterizes the type of the .raw file as binary. By defining the **ElementType** as $MET_UCHAR$, each item of the 3D tumor matrix is matched to a byte in the .raw file.

The **BinaryDataByteOrderMSB** and **CompressedData** tags define the endianness of the .raw file as little and its compression status as not compressed, respectively.

The **AnatomicalOrientation** tag describes the coordinate system as RAI and the **TransformMatrix**, **Offset** and **CenterOfRotation** tags describe the transformation that needs to be applied in order for the IJK coordinate system to be tranformed into the RAI coordinate system.



```
ObjectType = Image
NDims = 3
BinaryData = True
BinaryDataByteOrderMSB = False
CompressedData = False
TransformMatrix = <a> <b> <c> <d> <e> <f> <g> <h> <i>
Offset = <x_offset> <y_offset> <z_offset>
CenterOfRotation = <c_x> <c_y> <c_z
AnatomicalOrientation = RAI
ElementSpacing = <x_spacing> <y_spacing> <z_spacing>
DimSize = <x_dim> <y_dim> <z_dim>
ElementType = MET_UCHAR
ElementDataFile = data_file.raw
```

**Figure 3.2:** A standard format for the .mhd files that are given as input to the Nephroblastoma Oncosimulator. This format refers to a 3D image in the RAI coordinate system, which is described in a .raw binary file. Each element of the 3D image is matched to one byte of the .raw file. The .raw file is not compressed and is expressed in little endian.

Finally, the **ElementSpacing** tag describes the physical size of each item, i.e. of each geometrical cell or voxel of the 3D matrix tumor, in *mm* and for each dimension ($x\_spacing$ for the x dimension, $y\_spacing$ for the y dimension and $z\_spacing$ for the z dimension), while the **DimSize** tag describes the size of each dimension in geometrical cells, i.e. the number of items in each dimension ($x\_dim$ for the x dimension, $y\_dim$ for the y dimension and $z\_dim$ for the z dimension). The two subsequent reasonable conclusions are that:

- the physical size of the tumor region along each dimension is given as $i\_dim \cdot i\_spacing$, where $i = x, y, z$
- and the size of the .raw file in bytes is $x\_dim \cdot y\_dim \cdot z\_dim$.

### 3.2.2 A data preprocessing algorithm that explores the geometrical cell size for the optimal adaptation of the simulation resolution

As explained in **Chapter 2.3.3**, the size of the geometrical cell defines the resolution of the simulation. More specifically, as the size of the geometrical cell increases the simulation resolution decreases, since each geometrical cell carries a bigger number of actual biological cells and, as a result, each equivalence class inside each geometrical cell also carries an increased cell population. This means that more actual biological cells are simulated as one, i.e. they execute the same cell cycle transitions (see **Figure 2.8**) and at the same time points. Ideally, all the actual biological cells would be simulated separately from one another, i.e. they would be matched to a unique voxel of the 3D tumor matrix, which of course is not feasible or practical. However, despite the decrease in resolution, the increase of the geometrical cell dimension size reduces the cost of the simulation, with respect to the execution time, as well as to the required machine space. In fact, a certain decrease in the simulation resolution could be characterized as tolerable if the error that it causes is minimal and comparable to other errors that the deviation of the simulation from the actual physical phemomenon inevitably introduces. At the same time, the corresponding cost minimization could be benefactory for the execution.

This tradeoff between the simulation resolution and the simulation cost, which is tied to the value for the size of the geometrical cell, i.e. to the *i_spacing* values of **Figure 3.2** ($i = x, y, z$), raises the question of how to compute the optimal value for the geometrical cell size, so that the cost is minimized while the resolution is adequately maintained. This problem is addressed in the data preprocessing step of the workflow, since it can be solved independently for each patient input, in order to prepare them for the next steps of the workflow (see **Figure 3.1**). The simplistic algorithm that was implemented in order to explore this tradeoff problem is briefly presented in the flowchart of **Figure 3.3**.

The idea behind the algorithm of **Figure 3.3** is to explore the increase of the geometrical cell size for a specific input patient and within a tolerable error window. In order to do that, at first, the simulation is executed with the default patient specific size for the geometrical cell, as it is defined in the corresponding .mhd file, and the resulting outputs are recorded. Subsequently, the simulation is re-executed multiple times with a continuously growing scale factor for the geometrical cell size. The scale factor is incremented by a constant step, until the recorded outputs produce an error that exceeds the defined tolerable error window when compared to the outputs of the first default execution. Moreover, the algorithm takes into consideration the fact that if the input tumor files do not describe a tumor region with abundant normal tissue surrounding the tumor tissue, then, during the simulation and if the tumor tissue grows dramatically, it may hit the boundaries of the 3D matrix, which is defined statically and in turn causes the execution to error out with a suitable explanatory message. In order to address this issue, the algorithm uses one more input modifying parameter, a padding factor, which expresses the percentage by which the 3D matrix tumor should be padded with normal tissue, i.e. with zeros. This padding factor is also continuously incremented by a constant step until an errorless simulation execution for the currently explored scale factor is achieved.

The effect of the two input modifying factors, the scale and the padding factor, is formally and mathematically expressed in **Equation 3.1**. The scale factor defines the factor by which the geometrical cell size is to be multiplied, while the padding factor defines the percentage by which the 3D matrix tumor should be padded with normal tissue. The padding is performed first, increasing the size of the 3D tumor matrix, and then the scaling is performed, increasing the size of the geometrical cell and reducing the size of the 3D tumor matrix respectively. The new values for the geometrical cell size and the 3D matrix size are computed according to **Equation 3.1**.

$$\begin{aligned} i\_dim_{pad} &= (1+pad)\cdot i\_dim_{old} \\ i\_spacing_{new} &= scale \cdot i\_spacing_{old} \\ i\_dim_{new} &= \frac{i\_dim_{pad}}{scale} \\ i &= x, y, z \end{aligned} \tag{3.1}$$

At first, the *scale_step* and *pad_step* constant values are defined, expressing the loop incrementing step for each corresponding factor. The default value for the *scale_step* is 1, while the default value for the *pad_step* is 0.1. The *error_criterion* and the *error_window* constants are also defined at this time. The *error_criterion* expresses the standard by which the simulation error is computed. The default criterion is the relative tumor volume reduction, but any other simulation output parameter or a combination of a set of them can be utilized (see **Table 2.2**). The *error_window* expresses the tolerable error window and its default value is 0.1, meaning that the result of the simulation execution is allowed to deviate from that of the original simulation execution by 10%.

Then, after *pad* and *error* are initialized to 0 and *scale* is initialized to 1, the first and standard simulation is executed using the original provided patient specific input. If the execution errors out due to the 3D matrix boundaries being hit by the tumor, then the *pad* is incremented by *pad_step* and a new padded version of the patient input is created, with the appropriate modifications in the .raw file and the .mhd file according to **Equation 3.1**. This process is repeated until the *pad* is sufficient for the boundaries to not be hit and the patient input versions that are produced during this process are stored in the patient input DB. After the input has been padded sufficiently, the *error* is computed according to **Equation 3.2** and compared to *error_window*. If the *error* is greater than the *error_window*, then the algorithm terminates, while if it is lower than the *error_window*, then the *scale* is incremented by *scale_step* and a new scaled version of the patient input is created, with the appropriate modifications in the .raw file and the .mhd file according to **Equation 3.1**. The *pad* and *scale* variables act cooperatively as global variables in the double nested loop process that was described. During the first outer loop, with $scale = 1$, the error is equal to 0 and the *scale* is incremented to 2. The outer loop is repeated until the *scale* becomes so big that the resolution decreases to the point where the *error* becomes too big for the tolerable *error_window*. The inner loop is responsible for fitting the tumor in the 3D matrix for the entirety of the simulation and is repeated until the *pad* is big enough for the current scaled patient input to accomplish this goal.

$$error = \left| \frac{(\text{tumor volume reduction})_{current} - (\text{tumor volume reduction})_{original}}{(\text{tumor volume reduction})_{original}} \right| \tag{3.2}$$

The creation of the padded version of the input is trivial. However, there exists a variety of methods to scale the input, in order to produce a version with bigger geometrical cell size. These methods are based on combining the neighbor voxels of the 3D tumor, but they differ in the way that they decide the value of the resulting voxel. A simple approach is to use the mean or the median value of the neighboring voxels that are combined into the new voxel and if it surpasses a certain threshold, then the new voxel is categorized as tumor tissue (0xFF), else it is categorized as normal tissue (0x00). For instance, if the mean value is chosen with a threshold of 0.5, then the resulting voxel is considered as part of the tumor tissue only if at least half of the neighbor voxels that compose it belonged to the tumor tissue. The method that was finally utilized here is that of the spline interpolation.

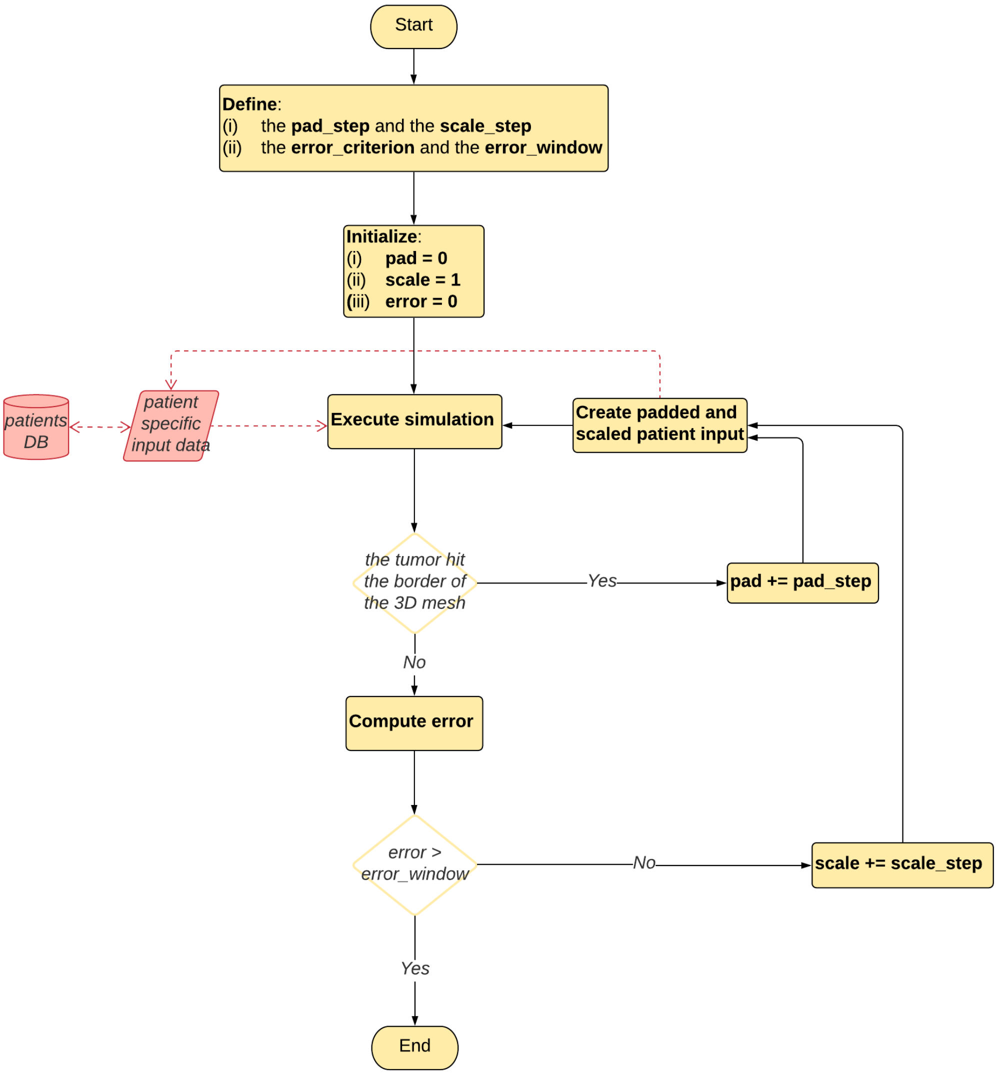

Start
Define:
(i) the pad_step and the scale_step
(ii) the error_criterion and the error_window
Initialize:
(i) pad = 0
(ii) scale = 1
(iii) error = 0
patients DB
patient specific input data
Execute simulation
Create padded and scaled patient input
the tumor hit the border of the 3D mesh
Yes
pad += pad_step
No
Compute error
error > error_window
No
scale += scale_step
Yes
End

**Figure 3.3:** The flowchart of the data preprocessing algorithm that explores the geometrical cell size for the optimal adaptation of the simulation resolution. The algorithm is comprised by a double nested loop. The inner loop determines the padding factor that ensures that the tumor will not hit the boundaries of the 3D matrix. The outer loop is responsible for increasing the scale of the geometrical cell, until the tolerable error window is transgressed.

## 3.3 Sensitivity Analysis

### 3.3.1 The role of sensitivity analysis as a method that is dynamically employed for the exploration of the Nephroblastoma Oncosimulator functionality under different parametric scenarios

Sensitivity analysis is a very useful tool for reviewing the funtionality of a given application, especially when its operations depend on a number of input parameters. The examination of the manner in which these parameters affect the output of the application, which is achieved via the sensitivity analysis, i.e. the process of exploring the range of values for the input parameter space and studying the variation that each change to the values for one or a combination of input parameters causes to the values for a set of output parameters, which are chosen to quantify the output of the application, is very useful in determining the behavior of the application under different parametric scenarios. In this context, sensitivity analysis is very helpful not only for gaining an insight on the application's functionality, but also in understanding the role and importance of each of the application's parameters. In this context, sensitivity analysis can be a strict, exact and well-defined tool for the verification of the application's robustness concerning the limitations that the range of values for its input parameters introduce and for extending and optimizing the application's results through the study and adaptation of its input parameters.

In the case of the Nephroblastoma Oncosimulator, sensitivity analysis is a very important tool with constant presence through many steps of the development and deployment of the model, from the definition, implementation and verification of its mechanisms to its clinical adaptation and clearance for clinical practice. The nature of the Oncosimulator software as a model that simulates the complex and multi-parametric phenomenon of cancer, renders it very fit for examinations and explorations that are subject to sensitivity analysis, not only during the early stages of its development, but also in the context of a continuous and dynamic process. The software's purpose, to offer personalized insight and feedback on the therapeutic scheme for a specific patient case, highlights the need for a constant exploration of the parameter space of the model, which is a process equivalent to the sensitivity analysis method, so that the optimal predictive results are ensured through the most fit adaptation of the parameters to the patient's image. Considering the possible impact of the Nephroblastoma Oncosimulator, beyond its immediate purpose as a medical advisor that suggests the optimal patient-specific therapeutic scheme, the exploration of its parameter space and the study of the corresponding effects on its outputs, along and during the process of its development, via dynamic sensitivity analysis, could have significant research value in the study of the complex phenomenon of cancer as a whole.

Here, the sensitivity analysis process is repeatedly employed, since it is applied for each optimized version of the Nephroblastoma Oncosimulator source code, according to the workflow outline of **Figure 3.1**. More specifically, the optimized versions of the model are compared to an original verified version, with concern to their behavior under various parametric scenarios. That way, any deviation in the behavior of the new optimized version from the behavior of the verified version is detected and handled. If no deviation is registered, then the new optimized version can be safely verified. Considering the importance of sensitivity analysis for the Nephroblastoma Oncosimulator, it has already been adequately performed for predecessor models in previous work. Different methods with varying foundation were utilized for the already performed sensitivity analysis, in order to ensure the validity of their results and the conclusive classification of the model parameters. The methods that stand out are three, a graphical method, a mathematical method and a statistical method. Here, only the graphical method was repeated and its results compared for each optimized versions to the ones for the original verified version.

### 3.3.2 Sensitivity analysis as a parametric exploration for each simulation input parameter separately and the parameters that are used for quantifying its results

The sensitivity analysis process, which is performed for the verification of each new optimized version of the Nephroblastoma Oncosimulator source code that is produced as an intermediate output of the workflow shown in **Figure 3.1**, is implemented via the exploration of the range of values for each input parameter separately. More specifically, a set of simulations is performed for each simulation input parameter, using the current source code version and with the parameter getting values from a predetermined exploration range, while the rest of the input parameters maintain a constant value that is equal to their reference value, as defined in **Table 2.7**.

The results of the sensitivity analysis are studied with respect to each input parameter separately via the comparison of the simulation outcomes for each set of simulations that are performed with the parameter getting values from a predetermined range while the rest maintain their values equal to their corresponding reference value. The outcomes of a single simulation are qunatified, with respect to the tumor volume reduction and the tumor's initial and final composition of the distinct cell type populations, using the simulation output parameters of **Table 2.2**.

These parameters are also utilized in order to express the outcomes of the set of simulations performed during the sensitivity analysis process. More specifically, the tumor size outcome of each simulation is quantified using the relative tumor volume variation percentage $DV$, computed according to **Equation 2.9**. The initial composition of the tumor is expressed according to the four initial population percentages of **Table 2.2**, one for each cell type, proliferative, dormant, differentiated and dead. For the final compostion of the tumor the relative population variation percentage for each distinct cell type is utilized, rather than the absolute value for the population percentages, as is the case for the initial composition. The relative population variation is computed for each cell type as the percentage of **Equation 3.3**. The set of parameters that are used in order to study the sensitivity analysis results are summarized in **Table 3.1**.

$$D_C = \frac{\frac{C_{final}}{\sum_C C_{final}} - \frac{C_{initial}}{\sum_C C_{initial}}}{\frac{C_{initial}}{\sum_C C_{initial}}} \tag{3.3}$$
$$C = PROLIF, DORMANT, DIFF, DEAD$$

| Symbol | Definition |
|---|---|
| $DV$ | relative volume variation percentage |
| $PROLIF_{initial}$ | initial percentage of proliferative cells |
| $DORMANT_{initial}$ | initial percentage of dormant cells |
| $DIFF_{initial}$ | initial percentage of differentiated cells |
| $DEAD_{initial}$ | initial percentage of dead cells |
| $D_{PROLIF}$ | relative variation of the percentage of proliferative cells |
| $D_{DORMANT}$ | relative variation of the percentage of dormant cells |
| $D_{DIFF}$ | relative variation of the percentage of differentiated cells |
| $D_{DEAD}$ | relative variation of the percentage of dead cells |

**Table 3.1:** The parameters that are utilized for the assessment of the sensitivity analysis results. They are derived from the simulation output parameters of **Table 2.2** and are used in order to monitor the effect that the variation in the values for a single input parameter has on the outcomes of the simulation, with respect to the tumor volume variation, the initial composition of the tumor and the corresponding relative cell population variation percentages for each distinct cell type, proliferative, dormant, differentiated and dead.

### 3.3.3 The graphical method for sensitivity analysis and its results for the original verified version of the model

The graphical method for sensitivity analysis is a simplistic, but accurate and thorough method that visualizes the relationship between the input and output parameters of the model in the form of scatter plots. More specifically, after the execution of the simulation with a range of values for each input parameter, the results are recorded in the form of the output parameters and the inferred relationship between them is visualized in the form of scatter plots.

This is a straightforward way to perform sensitivity analysis and express the results in a manner that highlights the intuition behind their understanding. This is the reason why this method was chosen to be explored and its results to be recreated here, since, due to its visual and intuitive nature, it enables the assessment of its results and their comparison to the ones acquired by previous work for the verified predecessor model in an easy and immediate manner.

The results of the graphical method for sensitivity analysis, as they are derived for the original verified version of the model and the corresponding source code, are displayed in **Figure 3.4** with respect to the effects of the variation of the values for each separate input parameter on the volume variation, in **Figure 3.5** with respect to the effects of the variation of the values for each separate input parameter on the initial composition of the tumor and in **Figure 3.6** with respect to the effects of the variation of the values for each separate input parameter on the variation of the tumor's composition.

Only the exploration of the input parameters that have a noticeable effect on the simulation outcomes are included. The tumor volume is affected by the $T_c$, $P_{sleep}$, $P_{sym}$, $R_A$, $P_{G0toG1}$, $T_{G0}$ and $CKR_{total}$ parameters. The tumor's initial composition is affected by the $P_{sleep}$, $P_{sym}$, $T_A$, $T_N$, $T_{G0}$, $R_{ADiff}$ and $R_{NDiff}$ parameters. Finally, the tumor's composition variation is affected by the $P_{sleep}$, $P_{sym}$, $T_{G0}$, $R_{ADiff}$, $R_{NDiff}$ and $CKR_{total}$ parameters.

Increase in the cell cycle duration $T_c$ causes the cell cycle to progress slower, which results in less mitoses and a bigger negative volume variation (volume reduction) for the constant time window of the simulation. However, no effects are recorded with respect to the tumor composition, both initial and final.

Increase in the ratio of cell that enter the dormant $G_0$ phase (i.e. increase in the value for the $P_{sleep}$ parameter) causes decrease in the percentage of proliferative cells in the initial tumor composition. The variation of the dead cells inside the tumor decreases, since the increased dormant cells in the initial tumor compositions do not perform many transitions in the cytokinetic diagram, which prevents the absolute number of cells from changing noticeably. This is also the reason why as the value for $P_{sleep}$ increases the tumor volume variation decreases and approximates zero.

Increase in the ratio of cells that divide symmetrically (i.e. increase in the value for the $P_{sym}$ parameter) causes decrease in the differentiated cell population of the initial tumor, since LIMP cells are produced with less frequence. As a result, a bigger number of divisions is performed, the absolute number of cells in the tumor increases and so does the tumor volume variation and the population of all the cell types except for the differentiated.

Increase in the maximum time spent in the $G_0$ phase befor the cell dies via necrosis (i.e. increase in the value for the $T_{G0}$ parameter) causes increase in the dormant cell population in the initial tumor, which in turn causes the tumor volume variation to approximate zero, since the dormant cells do not perform many transitions in the cytokinetic diagram and do not proliferate.

The cell kill ratio represents the effetcs of therpay and does not affect the initial composition of the tumor. Increase in the value for the $CKR_{total}$ parameter causes the therapy to hit more cells. As a result, the absolute number of cells in the tumor decreases and the tumor volume reduction increases.

Reviweing **Figures 3.4-3.6**, similar observations can be made for the rest of the input parameters that are explored via the graphical method sensitivity analysis. The main parameters that significantly affect the outcome of the simulation are the $P_{sleep}$, $P_{sym}$ and $CKR_{total}$ parameters. By performing the same method sensitivity analysis for each new version of the model and by comparing the results with the ones of **Figures 3.4-3.6**, any deviance of the new model behavior from the original model behavior can be monitored. That way, the new model can be corrected if needed and finally safely verified.

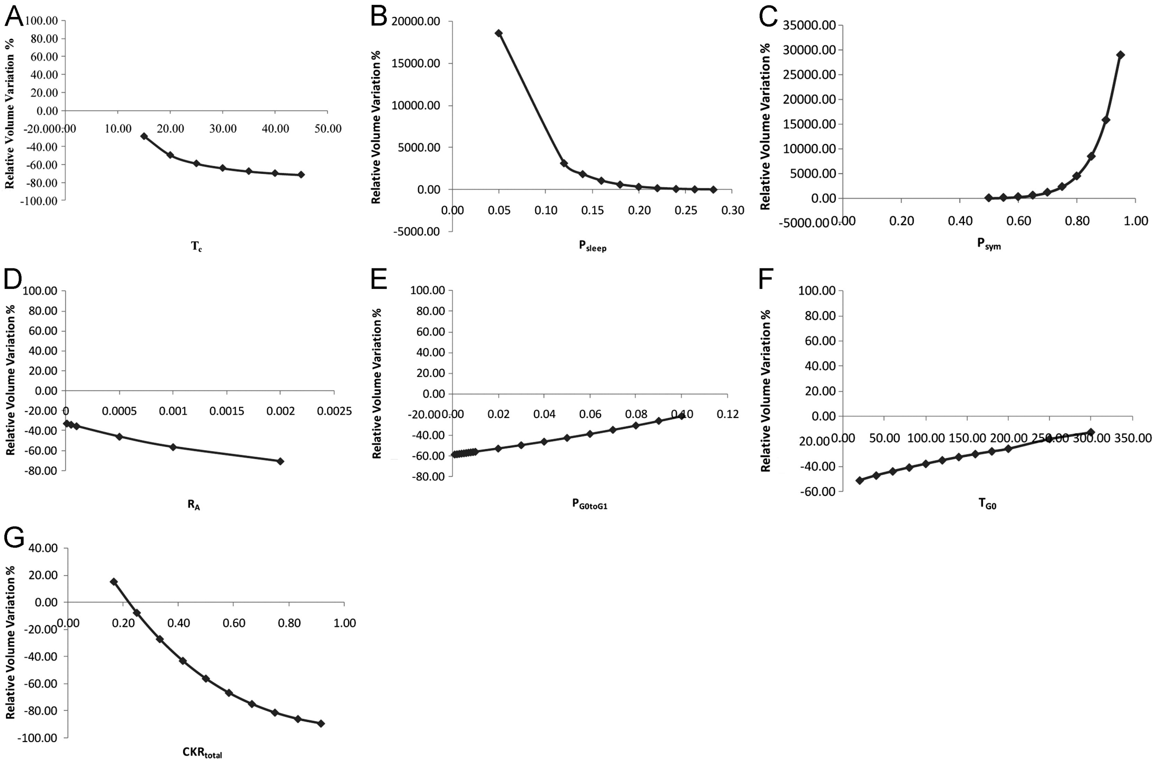


**Figure 3.4:** A visualization of the effect that the variation of the values for each input parameter has on the tumor volume variation, as it is derived via the graphical method for sensitivity analysis [71]. The tumor volume variation is measured by the relative tumor volume variation percentage (see **Equation 2.9**).

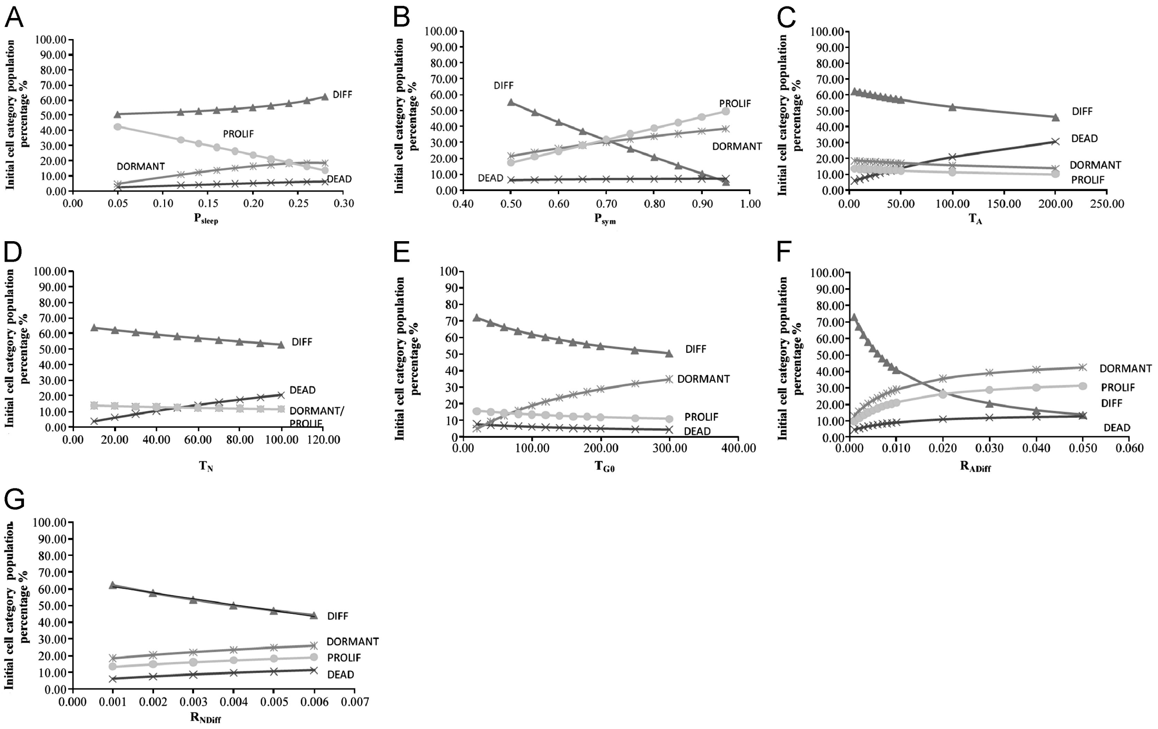


**Figure 3.5:** A visualization of the effect that the variation of the values for each input parameter has on the initial composition of the tumor, as it is derived via the graphical method for sensitivity analysis [71]. The initial tumor composition is measured according to the percentages of **Table 2.2**.

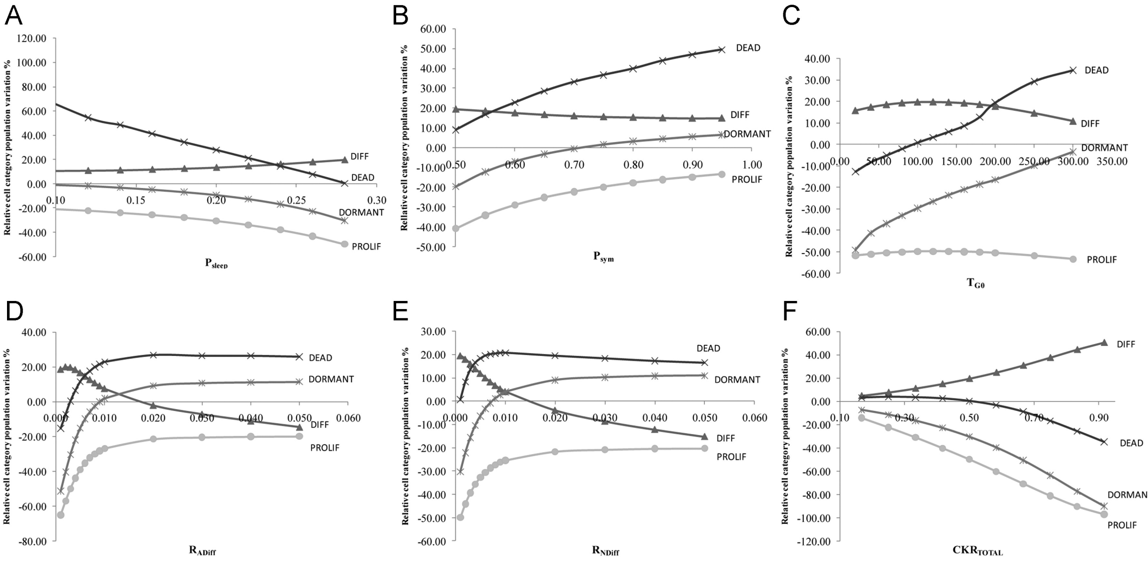


**Figure 3.6:** A visualization of the effect that the variation of the values for each input parameter has on the composition variation of the tumor, as it is derived via the graphical method for sensitivity analysis [71]. The tumor composition variation is measured by the relative population variation percentage, computed according to **Equation 3.3**.

## 3.4 Application Profiling & Code Optimization

### 3.4.1 The specifications of the architecture for the computing environment that was utilized

The series of simulations that were performed for the implementation of the workflow of **Figure 3.1** were executed on a linux server with a total of 48 CPUs. The detailed specifications of the computing environment were acquired using the `lscpu` linux command, whose output is displayed in **Table 3.2**.

The hardware of the computing environment includes two Intel(R) Xeon(R) CPU E5-2658A processors. This means that the architecture consists of 2 physical central processing units, each one encompassing 12 cores [72]. This can be also verified by checking the number of sockets and the number of cores per socket, as they are summarized in **Table 3.2**. The total number of physical cores is equal to the product of the number of physical CPUs with the number of cores per CPU, $2 \times 12 = 24$. Moreover, each core is matched to two threads (threads per core in **Table 3.2**), which doubles the 24 physical cores to 48 virtual cores. As a result, the server is assumed to operate using a total of 48 logical CPUs, which are numbered from 0 to 47. The matching of these logical CPUs to the physical cores of the harware is displayed in **Table 3.3**.

As for the memory architecture, the computing environment is equipped with a 3 level cache hierarchy. The first level of cache consists of 32 *KBytes* of memory for instruction storage and 32 *KBytes* of memory for general data storage. The second level of cache consists of 256 *KBytes* of memory and the third and final level consists of 30720 *KBytes* of memory.

The memory access is based on a NUMA (Non-Uniform Memory Access) architecture, which matches the 48 logical CPUs to 2 NUMA nodes, numbered from 0 to 1. The NUMA node 0 contains the $0 - 11$ and $24 - 35$ logical CPUs, while the NUMA node 1 contains the $12 - 23$ and $36 - 47$ logical CPUs. This matching is also evident in **Table 3.3**.

The information provided by the `lscpu` command are completed by some general data concerning the supported architecture (x86_64), CPU op-modes (32-bit and 64-bit) and byte order system (little endian), as well as the CPU model (63), family (6) and minimum and maximum frequency ($1200,0000$ $MHz$ and $2900,0000$ $MHz$ respectively).

```
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                48
On-line CPU(s) list:   0-47
Thread(s) per core:    2
Core(s) per socket:    12
Socket(s):             2
NUMA node(s):          2
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 63
Model name:            Intel(R) Xeon(R) CPU E5-2658A v3 @ 2.20GHz
Stepping:              2
CPU MHz:               1200.375
CPU max MHz:           2900,0000
CPU min MHz:           1200,0000
BogoMIPS:              4401.42
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              30720K
NUMA node0 CPU(s):     0-11,24-35
NUMA node1 CPU(s):     12-23,36-47
```

**Table 3.2:** The specification of the architecture of the linux server that was used for the simulation executions. These specifications were provided as the output of the `lscpu` command. The server consists of 2 processors, each one containing 12 physical cores (total number of physical cores equal to 24). Each core runs 2 threads (total number of virtual cores equal to 48). Moreover, the computing environment is equipped with a 3 level cache hierarchy. The memory access is based on a NUMA architecture, which consists of two nodes, one for each processor. Each one of the 24 physical cores is equipped with its own set of caches for the first two levels of cache hierarchy, while the last level of cache is shared between the physical cores that belong to the same processor (see **Table 3.3**).

| Logical CPU | Physical Core | Socket | NUMA Node | L1i | L1d | L2 | L3 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 2 | 2 | 0 | 0 | 2 | 2 | 2 | 0 |
| 3 | 3 | 0 | 0 | 3 | 3 | 3 | 0 |
| 4 | 4 | 0 | 0 | 4 | 4 | 4 | 0 |
| 5 | 5 | 0 | 0 | 5 | 5 | 5 | 0 |
| 6 | 6 | 0 | 0 | 6 | 6 | 6 | 0 |
| 7 | 7 | 0 | 0 | 7 | 7 | 7 | 0 |
| 8 | 8 | 0 | 0 | 8 | 8 | 8 | 0 |
| 9 | 9 | 0 | 0 | 9 | 9 | 9 | 0 |
| 10 | 10 | 0 | 0 | 10 | 10 | 10 | 0 |
| 11 | 11 | 0 | 0 | 11 | 11 | 11 | 0 |
| 12 | 12 | 1 | 1 | 12 | 12 | 12 | 1 |
| 13 | 13 | 1 | 1 | 13 | 13 | 13 | 1 |
| 14 | 14 | 1 | 1 | 14 | 14 | 14 | 1 |
| 15 | 15 | 1 | 1 | 15 | 15 | 15 | 1 |
| 16 | 16 | 1 | 1 | 16 | 16 | 16 | 1 |
| 17 | 17 | 1 | 1 | 17 | 17 | 17 | 1 |
| 18 | 18 | 1 | 1 | 18 | 18 | 18 | 1 |
| 19 | 19 | 1 | 1 | 19 | 19 | 19 | 1 |
| 20 | 20 | 1 | 1 | 20 | 20 | 20 | 1 |
| 21 | 21 | 1 | 1 | 21 | 21 | 21 | 1 |
| 22 | 22 | 1 | 1 | 22 | 22 | 22 | 1 |
| 23 | 23 | 1 | 1 | 23 | 23 | 23 | 1 |
| 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 25 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 26 | 2 | 0 | 0 | 2 | 2 | 2 | 0 |
| 27 | 3 | 0 | 0 | 3 | 3 | 3 | 0 |
| 28 | 4 | 0 | 0 | 4 | 4 | 4 | 0 |
| 29 | 5 | 0 | 0 | 5 | 5 | 5 | 0 |
| 30 | 6 | 0 | 0 | 6 | 6 | 6 | 0 |
| 31 | 7 | 0 | 0 | 7 | 7 | 7 | 0 |
| 32 | 8 | 0 | 0 | 8 | 8 | 8 | 0 |
| 33 | 9 | 0 | 0 | 9 | 9 | 9 | 0 |
| 34 | 10 | 0 | 0 | 10 | 10 | 10 | 0 |
| 35 | 11 | 0 | 0 | 11 | 11 | 11 | 0 |
| 36 | 12 | 1 | 1 | 12 | 12 | 12 | 1 |
| 37 | 13 | 1 | 1 | 13 | 13 | 13 | 1 |
| 38 | 14 | 1 | 1 | 14 | 14 | 14 | 1 |
| 39 | 15 | 1 | 1 | 15 | 15 | 15 | 1 |
| 40 | 16 | 1 | 1 | 16 | 16 | 16 | 1 |
| 41 | 17 | 1 | 1 | 17 | 17 | 17 | 1 |
| 42 | 18 | 1 | 1 | 18 | 18 | 18 | 1 |
| 43 | 19 | 1 | 1 | 19 | 19 | 19 | 1 |
| 44 | 20 | 1 | 1 | 20 | 20 | 20 | 1 |
| 45 | 21 | 1 | 1 | 21 | 21 | 21 | 1 |
| 46 | 22 | 1 | 1 | 22 | 22 | 22 | 1 |
| 47 | 23 | 1 | 1 | 23 | 23 | 23 | 1 |

**Table 3.3:** The mapping of the utilized architecture's logical CPUs (48) to the physical cores (24), NUMA nodes (2) and cache hierarchy (24 cache sets for the first and second level and 2 caches for the last level).

Reviewing the mappings of **Table 3.3**, it is evident that each one of the 24 physical cores that belong to the 2 distinct hardware processors of the architecture is equipped with its own set of caches for the first two levels of the cache hierarchy, while the last level of the cache hierarchy is shared between the cores that belong to the same NUMA node (and processor).

### 3.4.2 Valgrind as the profiling tool that was used for the estimation of the memory footprint per simulation execution and of the costs per simulation scan with concern to the CPU cycles and the memory accesses

Valgrind is an opensource software that is employed for the profiling of program executions on platforms that use unix-based operating systems [73]. The profiling of a program's execution proves useful in multiple scenarios, such as:

- in order to ensure that the program is free of bugs and that its behavior is the expected one
- in order to find where a suspected bug occurs and take action to eliminate it
- in order to check if the memory is managed properly or if memory leakage issues should be addressed
- in order to check the communication and synchronization between the various running threads in the case that multithreading is employed
- in order to study the program's performance, with respect to the costs that each distinct section of it introduces, in the attempt to optimize it

The valgrind framework includes a number of built-in tools that are used with the executable program to be profiled, regardless of the programming language of the corresponding source code. The suitable tool is chosen depending on the purpose of the profiling. Moreover, profiling tools can be custom-built on the valgrind framework, so that the purposes of the desired analysis are better accommodated.

Here, the `massif` tool was employed for the estimation of the memory footprint of each simulation execution and the `callgrind` tool was employed for the estimation of the cost of each simulation scan, with respect to the CPU cycles and the memory accesses.

The `massif` tool is a heap profiler that is provided by the valgrind framework [74]. It is used for the determination of the memory usage of the profiled program during its execution. More specifically, it periodically takes snapshots of the program's heap and it produces a graph showing the heap usage over the time duration of the program's execution. The snapshots are taken for every heap allocation/deallocation and when the maximum number of supported snapshots is reached, which is by default equal to 100 but can be adjusted via the `--max-snapshots` option, half of them are deleted. Most of the snapshots are normal, meaning that they are only depicted on the heap usage graph, but some are detailed, in that information concerning the program sections that allocated the heap memory is also provided for them. The detailed snapshots where the maximum heap memory is allocated are characterized as peak snapshots. Moreover, stack memory usage can be profiled too by setting the option `--stacks=yes`.

When programs are executed with the `massif` tool, they run about 20x slower than normal. In order for the time of the program execution with the `massif` profiler to be limited, the following methods can be employed:

- set `--stack=no` (default)
- reduce the frequency of detailed snapshots (`--detailed-freq` option, default value equal to 10)
- reduce the number of maximum snapshots (`--max-snapshots` option, default value equal to 100)
- reduce the maximum depth of the allocation trees recorded for detailed snapshots (`--depth` option, default value equal to 30)
- reduce the accuracy with which the peak snapshots are measured (`--peak-inaccuracy`, default value equal to 1.0)

Here, the `massif` heap profiler is utilized for the estimation of the memory footprint for each simulation execution. This is very important information, since it describes the memory needs of the execution, which in turn determines the number of executions than can be performed concurrently depending on the total available memory of the computing environment. The supported number of parallel simulation executions is a crucial information for the clinical adaptation step of the workflow, which is based on the execution of a number of independent simulations with varying values for the simulation input parameters.

The Nephroblastoma Oncosimulator executable allocates the memory that is necessary for the representation of the solid tumor in silico as a 3D matrix of geometrical cells, with each geometrical cell encompassing a number of equivalence classes, one for each phase of the cytokinetic diagram (see **Figure 2.9**). Assuming that the number and the types of the variables that describe the equivalence classes and the transitions between them for each geometrical cell are constant, then the memory required for each geometrical cell of the 3D matrix is alsoa known constant. This means that the total memory required by the executable depends on the number of geometrical cells in the 3D matrix, i.e. on the size of the 3D matrix, which is described in the files that summarize the medical imaging data of the patient and that are given as input to the executable.

The allocation of the memory for the 3D matrix happens during the first scan of the algorithm, which is responsible for the initialization of the 3D tumor mesh (see **Figure 2.7**). No major allocations and deallocations happen during the simulation evolution and the final memory deallocation happens with the completion of the execution. This means that the image of the heap usage graph does not change drastically during the simulation execution and that the desired memory footprint can be acquired using a minimal number of snapshots. Moreover, the memory is allocated on heap, which eliminates the need for the inclusion of stack profiling.

All in all, the `massif` tool was used for the estimation of the memory footprint of each simulation execution, which depends on the size of the patient-specific medical imaging input. For most inputs, the default tool options were used. For bigger inputs where the execution was too slow, the maximum number of snapshots was reduced, which is allowed due to the simple allocation pattern of the executable.

The `callgrind` tool is a profiler, provided by the valgrind framework and built on top of the `cachegrind` cache profiler, which documents the series of the events of the program's execution in the form of a call-graph [75]. More specifically, it records the number of instructions executed, their relationship to source lines, the caller/callee relationship between functions and the numbers of such calls.

The `callgrind` profiler can also optionally simulate a cache hierarchy and record the cache runtime behavior, counting the cache hit and miss events and attributing them appropriately to the responsible program source lines. This optional functionality is activated by setting the corresponding tool option ( `--cache-sim=yes` ). A similar option exists for the inclusion of branch prediction simulation and profiling ( `--branch-sim=yes` ).

Here, the `callgrind` profiler is employed for the atribution of the execution costs to the corresponding sections of the program's source code, with respect to both the CPU cycles and the memory accesses. This information is crucial for the process of locating the major cost inducing sections of the Nephroblastoma Oncosimulator source code. That way, the cost inducing factors can be studied and plans for their elimination and overall code optimization can be devised and implemented.

The simulation of the cache hierarchy, when the `--cahce-sim` option is set, is performed in the same manner as when the `cachegrind` tool is used [76]. Only the firt-level and last-level caches are simulated. The first level consists of two independent caches, one instruction cache ($I1$) and one data cache ($D1$). Its simulation is important since it masks the immediate interaction of the processor with the cache hierarchy. The last-level cache ($LL$) is unified and its simulation is important since it masks the immediate interaction of the cache hierarchy with the main memory. The recorded events include:

- instruction reads ($Ir$)
- instruction read misses ($I1mr$ for access to the first-level cache and $ILmr$ for access to the last-level cache)
- data reads ($Dr$)
- data read misses ($D1mr$ for access to the first-level cache and $DLmr$ for access to the last-level cache)
- data writes ($Dw$)
- data write misses ($D1mw$ for access to the first-level cache and $DLmw$ for access to the last-level cache)

The estimation of the total number of CPU cycles is performed according to **Equation 3.4**. The total number of misses is computed as the sum of instruction read misses, data read misses and data write misses, for the first-level cache ($L1m$) and the last-level cache ($LLm$) respectively. For the estimation of the total number CPU cycles, each instruction execution ($Ir$) is assumed to require one cycle, each access to the last-level cache (counted by the misses to the first-level cache $L1m$) is assumed to require 10 cycles and each access to the main memory (counted by the misses to the last-level cache $L1m$) is assumed to require 100 cycles.

$$\begin{aligned} Cycles &= Ir + 10 \cdot L1m + 100 \cdot LLm \\ L1m &= I1mr + D1mr + D1mw \\ LLm &= ILmr + DLmr + DLmw \end{aligned} \tag{3.4}$$

### 3.4.3 The plan for the optimization of the third scan of the Nephroblastoma Oncosimulator algorithm and its implementation using the OpenMP API

As described in detail in the dedicated chapters of the **Theoretical Framework**, the Nephroblastoma Oncosimulator algorithm uses the hypermatrix notation in order to represent the solid tumor in silico as a 3D matrix of geometrical cells. More specifically, the 3D matrix matches the physical size and shape of the tumor, as they are determined via medical imaging techniques, along the 3 spatial axes. Each item of the 3D matrix corresponds to a voxel of the image of the tumor, which is also referenced as geometrical cell, since it corresponds to a quantum in the 3D physical space. The bigger the size of the geometrical cell, the greater the physical space that it encompasses, which in turn reduces the size of the 3D matrix, i.e. the size of the simulation input.

Each geometrical cell encompasses a number of biological cells, which reside in the physical space that corresponds to it. Moreover, the biological cells inside each geometrical cell are further categorized in equivalence classes, according to the phase of the cytokinetic diagram in which they currently reside (see **Figure 2.9**). Each geometrical cell is fully described by the $x$, $y$, $z$ coordinates that define its position on the 3D physical space, while each equivalence class is fully defined by the geometrical cell in which it resides (i.e. by the $x$, $y$, $z$ coordinates in the 3D space) and the phase $p$ of the cytokinetic diagram that it represents (see **Equation 2.2**). Moreover, the time $t$ defines the time point of the simulation evolution, which corresponds to a unique image of the 3D matrix that represents the solid tumor in silico.

During the simulation evolution, the biological cells constantly transition between the equivalence classes inside the geometrical cell that they belong to, performing the transitions of the cytokinetic diagram **Figure 2.8** according to the values for the parameters of **Table 2.1** (see **Figure 2.10**). The role of the **third scan** of the algorithm is to implement these transitions. More specifically, each equivalence class is matched to a timer which decrements as the simulation time increments. When the timer reaches zero, the biological cells that reside in the equivalence class are moved to the other equivalence classes of the same geometrical cell, performing the transitions of the cytokinetic diagram with respect to the provided input parameter values.

As the biological cells transition between equivalence classes, the timers of the equivalence classes that accept new cells are adjusted. More specifically, since the biological cells of each equivalence class are simulated as one with respect to the transitions of the cytokinetic diagram, the timer that determines when the transitions starting from the equivalence class specific phase will take place is computed as the median value of the time remaining in that phase for each biological cell. The remaining time for the biological cells that were already in the equivalence class is equal to the continuously decrementing timer, while the remaining time for the biological cells that just entered the equivalence class is equal to the time duration of the equivalence class specific phase, as it is determined by the corresponding input parameters.

The third scan of the algorithm, which is preceded by the second scan that determines the percentage of cells that are hit by therapy according to the value for the $CKR$ parameter, concludes the simulation in the cellular level of biocomplexity for the current time point of the simulation by performing the transitions of the cytokinetic diagram. The computations that are performed during the third scan of the algorithm concern transitions between equivalence classes of the same geometrical cell. This means that the data dependencies are contained within the geometrical cells and that the computations for each item of the 3D matrix can be performed independently and, as a result, concurrently.

After the simulation is "summarized" on the cellular level, a "jump" is made to the tissue level, where the simulation for the current time steps continues with the fourth, fifth and sixth scan. These scans use the monte carlo modeling technique for the implementation of the exchange of biological cells between the geometrical cells with the goal of restoring the cell density distribution along the tumor and adjusting the tumor size and shape. As a result, the simulation on the tissue level of biocomplexity entails data dependencies between the geometrical cells that compose the 3D tumor. Moreover, the probabilistic nature of the modeling techniques that are employed for the simulation on the tissue level renders the management of the data dependencies complex and unpredictable. Here, the dependencies for the simulation on the tissue level are assumed to affect all the geometrical cells and, as a result, the simulation for the fourth, fifth and sixth scan of the algorithm is performed serially for all the items of the 3D matrix that represents the solid tumor in silico.

The "summarize and jump" method is repeated for each time step until the simulation is terminated. At each time step, the the second and the third scan (simulation on the cellular level) can be computed concurrently for each geometrical cell, while the following three scans (simulation on the tissue level) is computed serially for the set of geometrical cells. The evolution of the tumor in time is also computed serially, since the computations of a given time point require the results of the simulation in the previous time point, both for the cellular and for the tissue level.

The concurrent execution of the third scan of the algorithm for each item of the 3D matrix that represents the solid tumor in silico (i.e. for each geometrical cell of the simulation) is implemented using the `OpenMP` API, which is a portable and scalable model that provides a simple and flexible interface for developing parallel applications on platforms ranging from embedded systems and accelerator devices to multicore systems and shared-memory systems [77]. More specifically, the `OpenMP` API offers a set of compiler directives, runtime routines and environemt variables for the creation, synchronization and communication of multiple threads on a shared-memory environment. Here, the `#pragma omp parallel for collapse(3)` directive was utilized for the execution of the third scan of the algorithm with concurrent threads for the independent geometrical cell computations. The number of threads to be executed in parallel is specified by setting `OMP_NUM_THREADS` to the appropriate value.

### 3.4.4 Speedup as the metric for the performance enhancement outcomes and the limitations introduced by Amdahl's law

In the context of the present work, the performance of the simulation executions for the Nephroblastoma Oncosimulator algorithm is enhanced using the following two methods:

- decreasing the input size and the simulation resolution by increasing the size of the geometrical cell, according to the data preprocessing algorithm that addresses the simulation resolution and costs tradeoff problem (see **Figure 3.3**)
- using multiple threads and the `OpenMP` API to execute the third scan of the algorithm concurrently for the geometrical cells of the 3D matrix that represents the solid tumor in silico

The first data preprocessing method for performance enhancement adjusts the input of the simulation and, as a result, reduces both the termination time and the memory needs of the simulation execution. On the other hand, the second method, which runs the computations for the third scan of the algorithm for each element of the 3D matrix concurrently in multiple threads, focuses on the optimization of the source code of the Nephroblastoma Oncosimulator and aims to reduce the execution time of the simulation.

The metric that is used for the evaluation of the performance enhancement for the two methods that were employed, with respect to the execution time reduction, is the **relative speedup**. Typically, speedup compares the performance of two systems processing the same problem by measuring the improvement in speed of execution of a task on two similar architectures with different resources. Here, it is used in order to measure the factor by which the simulation execution time is reduced for each performance enhancement method. Practically, if the original execution time is $T_{original}$ and the execution time after the performance enhancement is $T_{enhanced}$, the relative speedup of the performance enhanced execution compared to the original execution is given by **Equation 3.5**. That way ,the speedup metric expresses the factor by which the execution is accelerated by the corresponding enhancement method. For instance, if $S = 2$ for an increased geometrical cell size, then the corresponding reduction to the simulation input size made the simulation execution two times faster.

$$S = \frac{T_{original}}{T_{enhanced}} \tag{3.5}$$

In the case that the performance enhancement method refers to code optimization using improved resources, e.g. by performing parallel computations when the data dependencies allow it, the theoretical speedup is given by **Amdahl's law**.

Amdahl's law is based on the premise that each task can be slit in two parts, a part $(1-p)$ that does not benefit from the improved resources and a part $p$ that does, and that the theoretical speedup for the workload that uses improved resources compared to the initial workload is limited by the task portion that does not benefit from them.

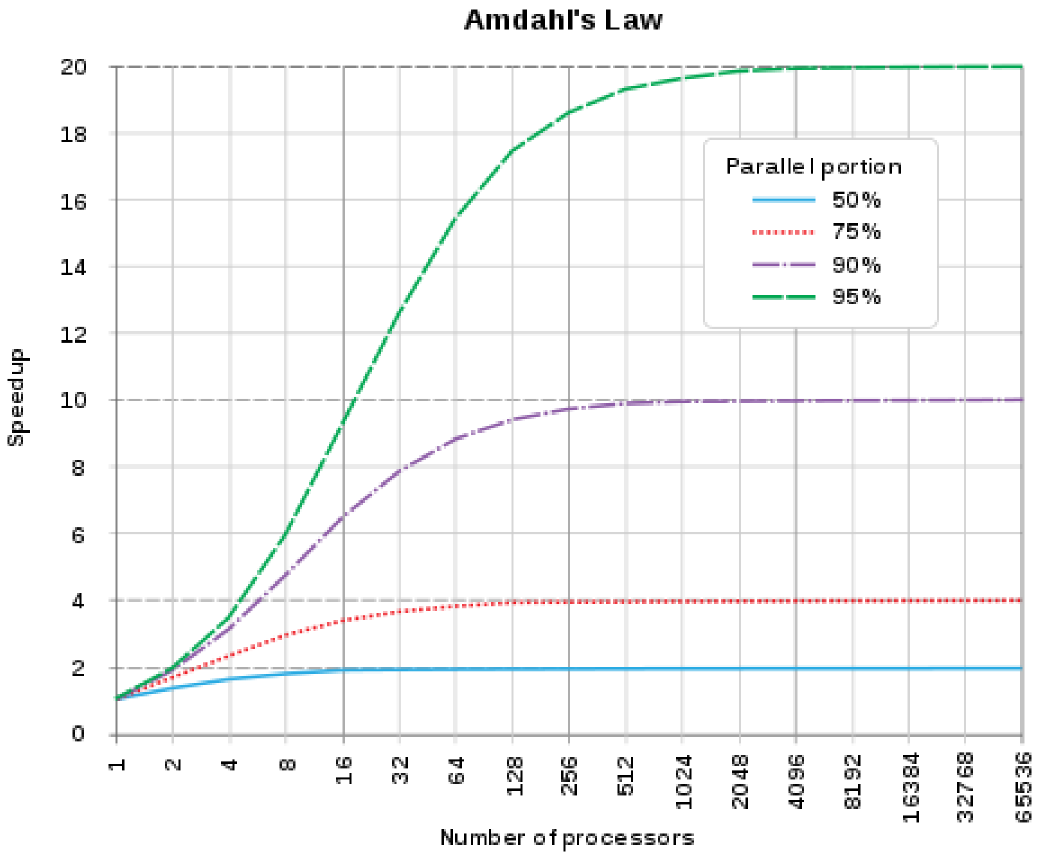


**Figure 3.7:** A visualization of the speedup of a workload as the number of utilized processors increases, according to Amdahl's law [78]. As the number of processors increases, so does the speedup, until it reaches the maximum speedup, as it is estimated in **Equation 3.7**. The maximum speedup is determined by the portion $p$ of the workload that benefits from the improved resources (parallel portion). As $p$ increases, so does the maximum speedup.

More specifically, if the total time for the original execution of the task, with no improved resources, is equal to $T$, then the task portion that does not benefit from the improved resources requires $(1-p) \cdot T$ time, while the task portion that benefits from the improved resources requires $p \cdot T$ time. When the improved resources are introduced, the part that benefits from them is accelerated with speedup $s$, while the part that does not benefit from them is not accelerated. Thus, the total execution time for the workload that uses the improved resources is:

$$T(s) = (1-p) \cdot T + \frac{p}{s} \cdot T$$

The formula that expresses Amdahl's law is given in **Equation 3.6** and is derived using **Equation 3.5** with $T_{original} = T$ and $T_{enhanced} = T(s)$.

$$S(s) = \frac{1}{(1-p) + \frac{p}{s}} \tag{3.6}$$

Indeed, the maximum speedup is limited according to **Equation 3.7**. As the portion $(1-p)$ of the task that does not benefit from the improved resources increases, the maximum theoretical speedup is decreased. This is in comliance with the intuition and understanding of the problem, since the part that does not utilize the improved resources is not accelerated. Consequently, when it takes up a big part of the execution time in the original system, the speedup potential is limited.

$$\begin{aligned} S(s) &\leq \frac{1}{1-p} \\ \lim_{s \to \infty} S(s) &= \frac{1}{1-p} \end{aligned} \tag{3.7}$$

## 3.5 Clinical Adaptation

### 3.5.1 The concept of virtual patients for the in silico representation of a single physical patient with a variety of values for the joint distribution of the simulation input parameters

The term virtual patient or avatar refers to the digitized simulated version of the image of an actual physical patient in silico. A virtual patient basically encompasses all the information of one distinct and full simulation, its input files and parameters, its evolution and its outcome and output parameters. Since the input files of the Oncosimulator are patient specific, it is obvious that each virtual patient is matched to one unique physical patient. However, the opposite is not true. A single physical patient can be matched to a multitude of virtual patients. That is because a virtual patient is defined, among others, by the values for the parameters of the simulation, input and output, as they are defined in **Tables 2.1 and 2.2**.

The input and output parameters of the simulation define a set of variables that, even though they have a degree of interdependence, create a joint parameter exploration space, which is immediately linked with the number of virtual patients for each physical patient. More specifically, each unique joint value distribution that is assigned to the parameters set corresponds to one unique virtual patient. Despite the interdependence between the simulation parameters and the limitations that the allowed values domain for each parameter introduces, it is clear that the combinations of values for the various parameters are so many that each physical patient can be matched to a practically indefinitely large set of virtual patients.

This property that matches one single physical patient to a multitude of virtual patients compliments the intuition behind the role of the simulation parameters and the function of the Oncosimulator as a patient specific application. Most of the simulation parameters define and drive the evolution of a series of biological processes. These processes, despite being determined by a set of natural and physical rules, are characterized by a significant heterogeneity among the patient population. For instance, the $CKR$ input parameter determines the effects of therapy, which can vary significantly between patients. Moreover, the histologic profile of the patient may categorize the neoplasm as aggressive or regressive, which is mirrored in the values for the symmetric division rate of the tumor cells and for the rate with which the tumor cells enter the dormant $G0$ phase, $Psym$ and $Psleep$ respectively.

This means that not all patients are succesfully simulated in silico and subsequently characterized by the same set of values for the simulation parameters. Moreover, the values for the simulation parameters cannot be inferred unambiguously and accurately from the patient specific medical data. Subsequently, the nature of the Oncosimulator as a patient specific application requires the study of the patients in the general context of virtual patients. By simulating the patient in silico for a multitude of joint distribution values assigned to the set of simulation parameters, i.e. matching the actual patient with multiple virtual patients, the adequate exploration of the parameter space is ensured. This principle is utilized in the clinical adaptation algorithm that is presented in the following section.

### 3.5.2 The clinical adaptation algorithm that explores the CKR parameter for each virtual patient with the goal of the optimal adaptation to the medical data of the physical patient

The matching of distinct physical patients to a multitude of virtual patients is utilized in the clinical adaptation process, in order to manipulate the simulation parameters in a manner that will lead the simulation towards a direction that is closer to the real evolution and outcome of the actual physical phenomenon. In this section, the clinical adaptation process refers to the execution of multiple simulations for patients that have already been treated and their outcomes and therapy results have been studied and recorded thoroughly, with the goal of determining the parameter value distribution that better aligns the simulation results with the actual recorded results. The flowchart of the implemented algorithm is shown in **Figure 3.8**.

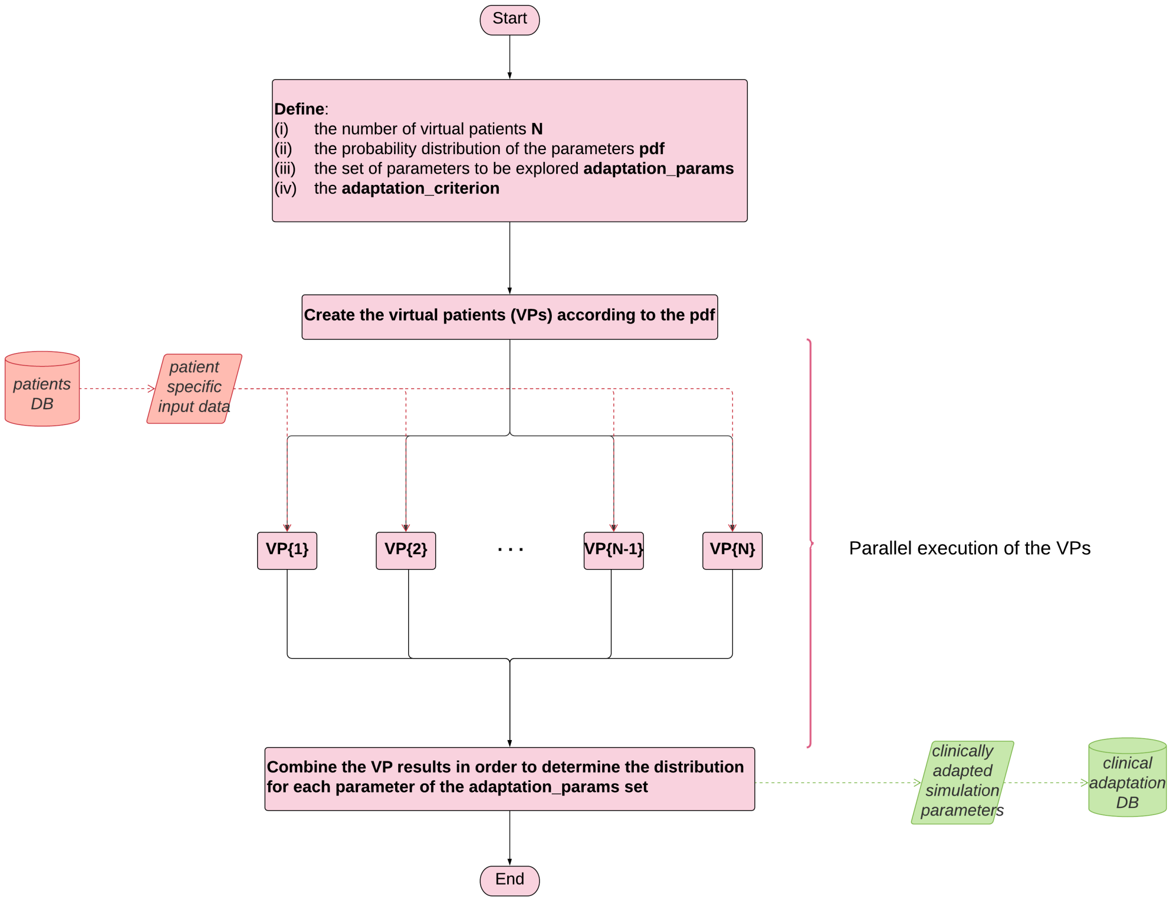

Start
Define:
(i) the number of virtual patients N
(ii) the probability distribution of the parameters pdf
(iii) the set of parameters to be explored adaptation_params
(iv) the adaptation_criterion
Create the virtual patients (VPs) according to the pdf
patients DB
patient specific input data
VP{1}
VP{2}
...
VP{N-1}
VP{N}
Parallel execution of the VPs
Combine the VP results in order to determine the distribution for each parameter of the adaptation_params set
clinically adapted simulation parameters
clinical adaptation DB
End

**Figure 3.8:** The flowchart of the clinical adaptation algorithm. Each actual physical patient is matched to $N$ virtual patients, i.e. $N$ distinct joint distribution values for the simulation parameters set. Here, $N = 20$ and the parameters are assigned their values from the uniform distribution with mean value the corresponding parameter reference value and a deviation of 50% (default *pdf*). The set of adaptation parameters (*adaptation_params*) is adjusted for each virtual patient, via a dedicated exploration algorithm, so that the parameter values that optimze the outcome of the simulation with respect to the degree that it approximates the medical outcome of the actual patient are determined. This adaptation is performed with respect to the *adaptation_criterion* (here the percentage of the therapy induced tumor volume reduction) and is independent for each distinct virtual patient, which allows the parallel execution of the virtual patients. Finally, the virtual patient execution results are evaluated and combined to produce the distribution of the adaptation parameters that is fit for the in silico patient representation.

At first, the total number of virtual patients $N$ for each physical patient is defined. The default value for $N$ was chosen to be 20. Moreover, the simulation parameters are separated into the set of parameters to be explored, whose distribution is determined as the output of the clinical adaptation, and the set of parameters whose distribution is predetermined. The parameters to be explored are kept in the *adaptation_params* set, while the probability distribution of the rest of the parameters is defined by *pdf*. The default *adaptation_params* set contains only the $CKR$ input parameter, but more complex explorations can be performed at a later time. The rest of the parameters are described by the uniform distribution (see **Figure 3.9**), which is the default value for the *pdf*. A more appropriate distribution could be the gaussian. However, the mean value and the standard deviation of the simulation parameters are not represented concisely in the bibliography. As a result, the unifrom distribution was utilized, with mean value the reference value for each parameter and the values for $a$ and $b$ as computed in **Equation 3.8**. The default value for the *window_ratio* was chosen to be 0.5, meaning that each parameter (except for the ones in the *adaptation_params* set) takes uniformly random values that are at most 50% higher or lower from the corresponding reference value. Finally, the *adaptation_criterion* with respect to which the exploration of the *adaptation_params* is performed for each virtual patient is defined. Here, the exploration of the *adaptation_params* is performed with respect to the therapy induced tumor volume reduction, until the in silico tumor volume reduction deviates from the actual tumor volume reduction by a percentage lower than 5%.

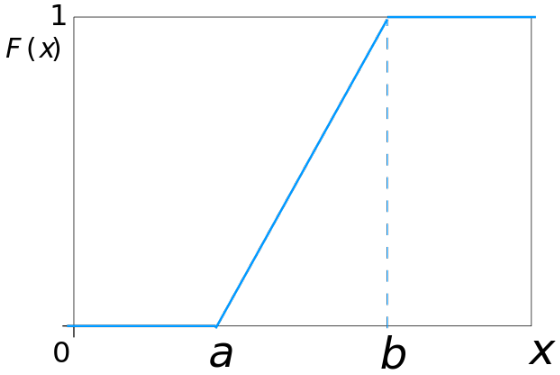


**Figure 3.9:** The probability density function for the uniform distribution [79]

.

$$\begin{aligned} a &= (1 - window\_ratio) \cdot mean \\ b &= (1 + window\_ratio) \cdot mean \end{aligned} \tag{3.8}$$

After those necessary definitions, $N$ distinct virtual patients are created with parameter values that are chosen according to the *pdf* that was defined, here according to the uniform distribution. Then, each virtual patient is simulated multiple times with a variety of values for the *adaptation_params*, which are methodically explored, in order for the optimal parameter adjustment to be accomplished, with respect to the *adaptation_criterion*. Note that the virtual patients are being executed in parallel, but the exploration of the values for the adaptation parameters for each virtual patient may be serial, depending on the implemented exploration algorithm. Here, only the $CKR$ parameter is explored, with the algorithm of **Figure 3.10**, which is by definition serial.

The implemented exploration of the $CKR$ parameter for each virtual patient, as it is described in the algorithm of **Figure 3.10**, is based on the fact that if the in silico tumor volume reduction is greater than the expected one, then the effects of the therapy are overestimated in the executed simulation and the value for the $CKR$ parameter needs to be decreased. On the other hand, if the in silico tumor volume reduction is lower than the expected one, then the effects of the therapy are underestimated in the executed simulation and the value for the $CKR$ parameter needs to be increased.

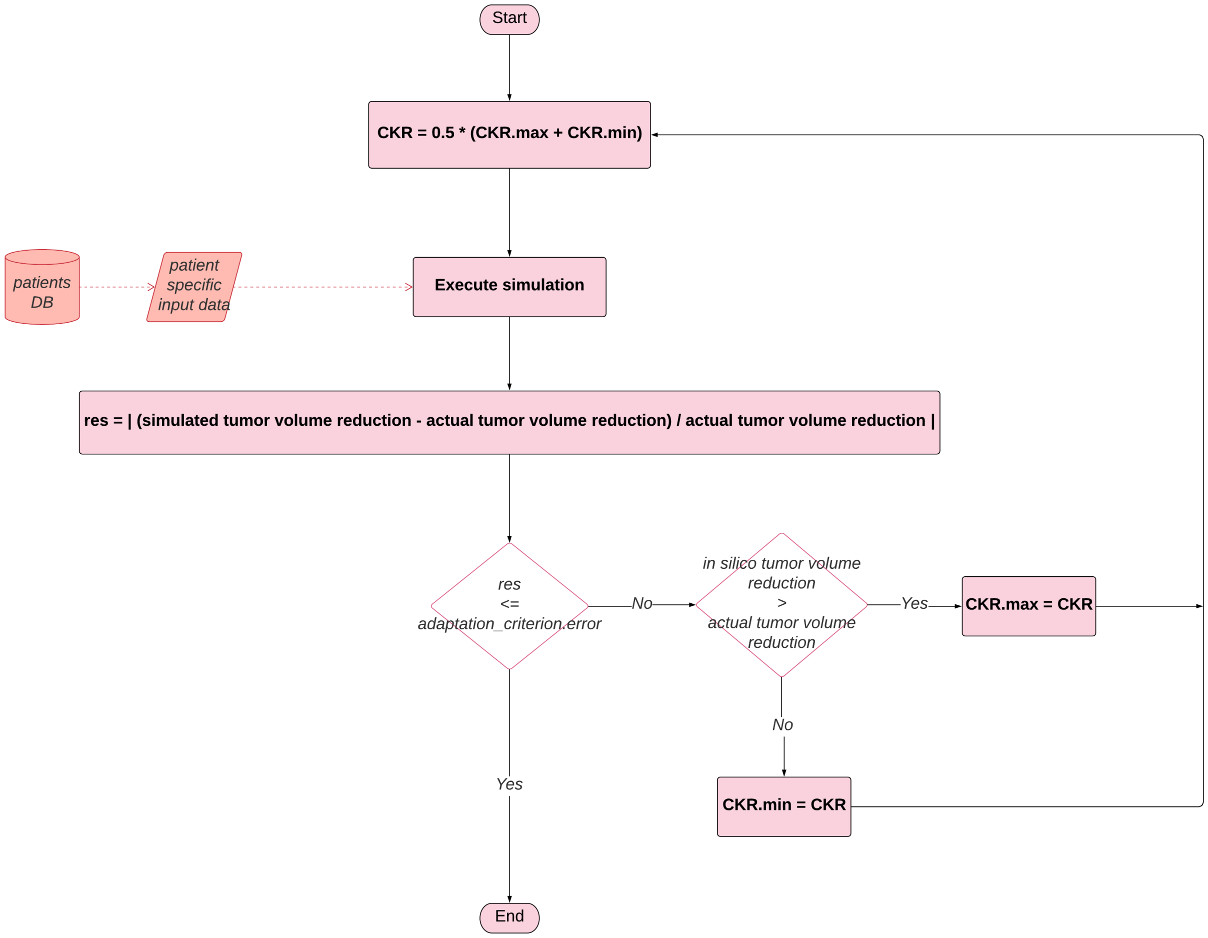

Start
CKR = 0.5 * (CKR.max + CKR.min)
patients DB
patient specific input data
Execute simulation
res = | (simulated tumor volume reduction - actual tumor volume reduction) / actual tumor volume reduction |
res <= adaptation_criterion.error
No
in silico tumor volume reduction > actual tumor volume reduction
Yes
CKR.max = CKR
No
CKR.min = CKR
Yes
End

**Figure 3.10:** The flowchart of the algorithm that explores the $CKR$ parameter value range for the optimal clinical adaptation. A simulation is executed with the $CKR$ parameter value equal to the median of the currently explored value range, which is defined by the corresponding minimum and maximum values ($CKR.min$ and $CKR.max$ respectively). The simulation error is estimated and compared to the tolerable error, with respect to the $adaptation\_criterion$. If the effects of the therapy are overestimated, then the desired value for the $CKR$ parameter is lower than the value that was just tested and $CKR.max$ is adjusted accordingly. If the effects of the therapy are underestimated, then the desired value for the $CKR$ parameter is greater than the value that was just tested and $CKR.min$ is adjusted accordingly. This process is repeated and the explored range is continuously shrinked until a sufficient value for the $CKR$ parameter is found. Since the $CKR$ parameter represents a probability, its initial minimum and maximum values are equal to 0 and 1 respectively.

The algorithm explores the value range for the $CKR$ parameter by referencing and altering its minimum and maximum value ($CKR.min$ and $CKR.max$ respectively). More specifically, a simulation is executed with the $CKR$ parameter value equal to the median of the value range that is being currently explored ($\frac{CKR.min+CKR.max}{2}$). Then, the error of the simulation is evaluated and compared to the tolerable error with respect to the $adaptation\_criterion$. Here, the error metric is the deviation percentage of the in silico tumor volume reduction from the actual tumor volume reduction and the allowed error is 5%. If the simulation error is lower than this percentage, then the optimal $CKR$ has been found. However, if the simulation error surpasses the allowed percentage, the $CKR$ exploration value range needs to be adjusted. More specifically, if the tumor volume reduction is overestimated (in silico tumor volume reduction > actual tumor volume reduction), then the optimal value for the $CKR$ parameter is lower than the one that was just tested and the $CKR.max$ is set to that value. On the other hand, if the tumor volume reduction is underestimated (in silico tumor volume reduction < actual tumor volume reduction), then the optimal value for the $CKR$ parameter is greater than the one that was just tested and the $CKR.min$ is set to that value. That way, the explored value range for the $CKR$ parameter is dynamically adjusted and shrinked, until an adequate value for the parameter is found. Since the $CKR$ parameter represents a probability, its initial minimum and maximum values are equal to 0 and 1 respectively.

After the set of *adaptation_params* has been explored for each virtual patient, each parameter that belongs to it (only the $CKR$ parameter here) is assigned a value that corresponds to an optimal simulation that leads to a minimal error with respect to the *adaptation_criterion* for each virtual patient. As the final step, the results for each virtual patient, including the values for the *adaptation_params* and the value for the joint distribution of the rest of the simulation parameters, are co-evaluated, in order for the probability distribution of the *adaptation_params* that corresponds to the actual patient to be determined.

# Results

## 4.1 Data Preprocessing

### 4.1.1 The original values for the parameters that are inherent to the simulation input files for each dataset patient

The primary parameters that are inherent to the input files of the simulation are responsible for describing the morphology and the size of the initial tumor, i.e. of the tumor before the preoperative chemotherapy treatment is applied, as it is depicted in the MRT that was acquired at the dates specified in **Table 2.9**. More specifically, these parameters consist of:

- the **x-dim**, **y-dim** and **z-dim** parameters, which specify the size of the 3D matrix that represents the initial tumor in silico, along each of the three axes,
- the **GC-dimension**, which specifies the physical size (in mm) that corresponds to the edge of each item (geometrical cell) of the 3D matrix that represents the initial tumor in silico,
- the **scale** and the **padding** factors, which are referenced in the resolution adaptation algorithm of **Figure 3.3** and represent the factor by which the geometrical cell size is multiplied and the percentage by which the tumor is padded with normal tissue (i.e. with zeros) along each axis.

In order to properly process the results of the resolution adaptation algorithm of **Figure 3.3**, it is essential that the values for these input file inherent primary parameters that describe the morphology and size of the initial tumor are known for each dataset patient. These parameter values for each set of original input files of the three dataset patients are summarized in **Table 4.1**.

The **scale** and **padding** factors are set to the identity element for their corresponding function, 1 and 0 respectively, since in the original input files no scaling of the geometrical cell dimension or padding of the 3D tumor is applied.

The **GC-dimension** is equal to 1 for all three of the dataset patients, which corresponds to the highest simulation resolution. The **x-dim**, **y-dim** and **z-dim** parameters are linked to the size of the tumor and get higher as the tumor volume increases.

The **3D Tumor Volume** expresses the physical volume that corresponds to the area of the 3D matrix that represents in silico the tumor along with its normal tissue microenvironment. It is computed (in mm) as (x-dim·y-dim·z-dim)·(GC-dimension)$^3$ and its difference percentage from the actual tumor volume, as estimated before the preoperative therapy via MRT (see **Tables 2.9 and 2.11**), expresses the **Padded**

**Space**, i.e. a rough estimate of the percentage by which the 3D tumor matrix is alraedy padded with normal tissue geometrical cells.

| **Pseudonym** | Patient_15 | Patient_16 | Patient_17 |
|---|---|---|---|
| **scale** | 1 | 1 | 1 |
| **padding** | 0 | 0 | 0 |
| **GC-dimension** | 1 | 1 | 1 |
| **x-dim** | 87 | 88 | 118 |
| **y-dim** | 68 | 94 | 123 |
| **z-dim** | 75 | 98 | 132 |
| **3D Tumor Volume** ($mm^3$) | 443700 | 810656 | 1915848 |
| **Image Pre-Chemo (Database MRT)** ($mm^3$) | 143987.13 | 306985.833 | 536733.98 |
| **Padded Space** (*percentage*) | 208.15 | 164.07 | 256.95 |

**Table 4.1:** The values of the primary input file inherent parameters for the original input files for each of the dataset patients. The scale and padding factors are set to the identity element for their corresponding function. The simulation is executed in the highest possible resolution (GC-dimension=1). The volume of the 3D tumor matrix ((x-dim · y-dim · z-dim) · (GC-dimension)$^3$ · $mm^3$) is bigger than the actual physical tumor volume, indicating the percentage by which the 3D tumor matrix is already padded with normal tissue geometrical cells (highest for Patient_17, medium for Patient_15 and lowest for Patient_16).

### 4.1.2 The results of the simulation resolution adaptation with respect to the execution time and error that corresponds to each scale and padding factor set for each dataset patient

The tolerable error window that was used during the execution of the resolution adaptation algorithm of **Figure 3.3**, which allows for the tumor volume reduction of the simulation to deviate from the one estimated by the original simulation with the parameters of **Table 4.1** by a percentage of 10%, leads to the exploration of a great range of values for the scale parameter of the algorithm, which exceeded the value of 50.

However, the typical values for the GC-dimension parameter do not surpass the value of 4, since the morphology of the tumor may be distorted for greater values, even if the tumor volume reduction criterion of the resolution adaptation algorithm indicates a good adaptation with a minor error percentage. That is why here only the exploration of scale and padding factor values that lead to simulations with GC-dimension values that do not surpass the threshold of 10 were taken into account.

The results of the simulation resolution adaptation algorithm for each dataset patient are presented in **Tables 4.2, 4.3 and 4.4**. Only the simulation executions that were completed without error are recorded. Each recorded execution corresponds to a distinct scale factor, along with the padding factor that ensures that the tumor will not hit the boundaries of the 3D matrix during the simulation evolution and that the simulation will not error out. The corresponding GC-dimension, x-dim, y-dim and z-dim parameter values are in comliance with **Equation 3.1**. Note that as the scale factor increases, the padded space of the 3D tumor decreases, which may cause the simulation to error out due to the trasgression of the 3D matrix boundaries. In that case, the padding factor is appropriately adjusted by the inner loop of the algorithm and the padded space becomes fit for the normal execution of the simulation again. The corresponding total simulation execution time, as well as the tumor volume reduction error, are also recorded. As the scale factor increases, the execution time of the simulation decreases, as expected, since the input size is reduced. On the other hand, the tumor volume reduction error fluctuates with the increase of the scale factor, a fact that expresses the tradeoff that was explained in **Chapter 3.2.2**.

The relationship between the scale factor, the simulation execution time and the tumor volume reduction error is visualized for each dataset patient in **Figures 4.1, 4.2 and 4.3**, in the form of scatter plots. The execution time of the simulation, which is measured in sec, is also analyzed with respect to the distinct scans of the simulation (see **Figure 2.7**), according to the time that each scan occupies. These times, as well as the tumor volume reduction error, are represented on the y axis of the scatter plots, while the scale factor is represented in the x axis of the scatter plots. Moreover, the padding factor value that corresponds to each scale factor value is denoted by the color of the scatter points, according to the depicted colorscale.

The most dramatic decrease in the simulation times, regarding both the total execution time, as well as the time that each separate scan occupies, is recorded for the increase of the scale factor value from 1 to 2. With the increase of the scale factor value from 2 to 3, only the first three simulation scans become noticeably faster, while for further increase of the scale factor value, the decrease in the simulation times is insignificant.

Moreover, the scans that occupy most of the execution time are the three last scans (fourth, fifth and sixth), which comprise the biomechanical part of the simulation. These are the scans that cause an increase of the total simulation execution time that can be problematic, since it cannot be easily addressed due to the existing data dependencies. That is why the data preprocessing step should be focused in decreasing the time that these scans require. Reviewing **Figures 4.1-4.3**, it becomes evident that the a scale factor of 2 adequately reduces the times of these scans, as well as the total simulation execution time. In any case, a scale factor greater than 3 does not need to be considered.

| **scale** | **padding** | **GC-dimension** | **x-dim** | **y-dim** | **z-dim** | **Total simulation execution time** (*sec*) | **Tumor Volume Reduction Error** (*percentage*) | **3D Tumor Volume** ($mm^3$) | **Padded Space** (*percentage*) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.1 | 1 | 105 | 82 | 91 | 1777.56 | 0.00 | 783510 | 444.15 |
| 2 | 0.1 | 2 | 52 | 41 | 46 | 47.77 | 0.04 | 784576 | 444.89 |
| 3 | 0.1 | 3 | 35 | 27 | 30 | 7.90 | -0.02 | 765450 | 431.61 |
| 4 | 0.2 | 4 | 31 | 24 | 26 | 4.48 | 0.00 | 1238016 | 759.81 |
| 5 | 0.2 | 5 | 25 | 19 | 21 | 2.06 | 0.20 | 1246875 | 765.96 |
| 6 | 0.2 | 6 | 20 | 16 | 18 | 1.25 | 0.04 | 1244160 | 764.08 |
| 7 | 0.2 | 7 | 18 | 14 | 15 | 0.89 | -0.04 | 1296540 | 800.46 |
| 8 | 0.2 | 8 | 15 | 12 | 13 | 0.60 | -0.04 | 1198080 | 732.07 |
| 9 | 0.2 | 9 | 14 | 11 | 12 | 0.43 | 0.07 | 1347192 | 835.63 |
| 10 | 0.3 | 10 | 14 | 11 | 12 | 0.41 | 0.09 | 1848000 | 1183.45 |

**Table 4.2:** The results of the resolution adaptation algorithm for Patient_15.
The recorded scale factor values range from 1 to 10. Each scale factor value corresponds to a padding factor value that ensures the normal simulation execution. The tumor volume reduction error fluctuates with the scale factor value increase, while the total simulation execution time decreases.

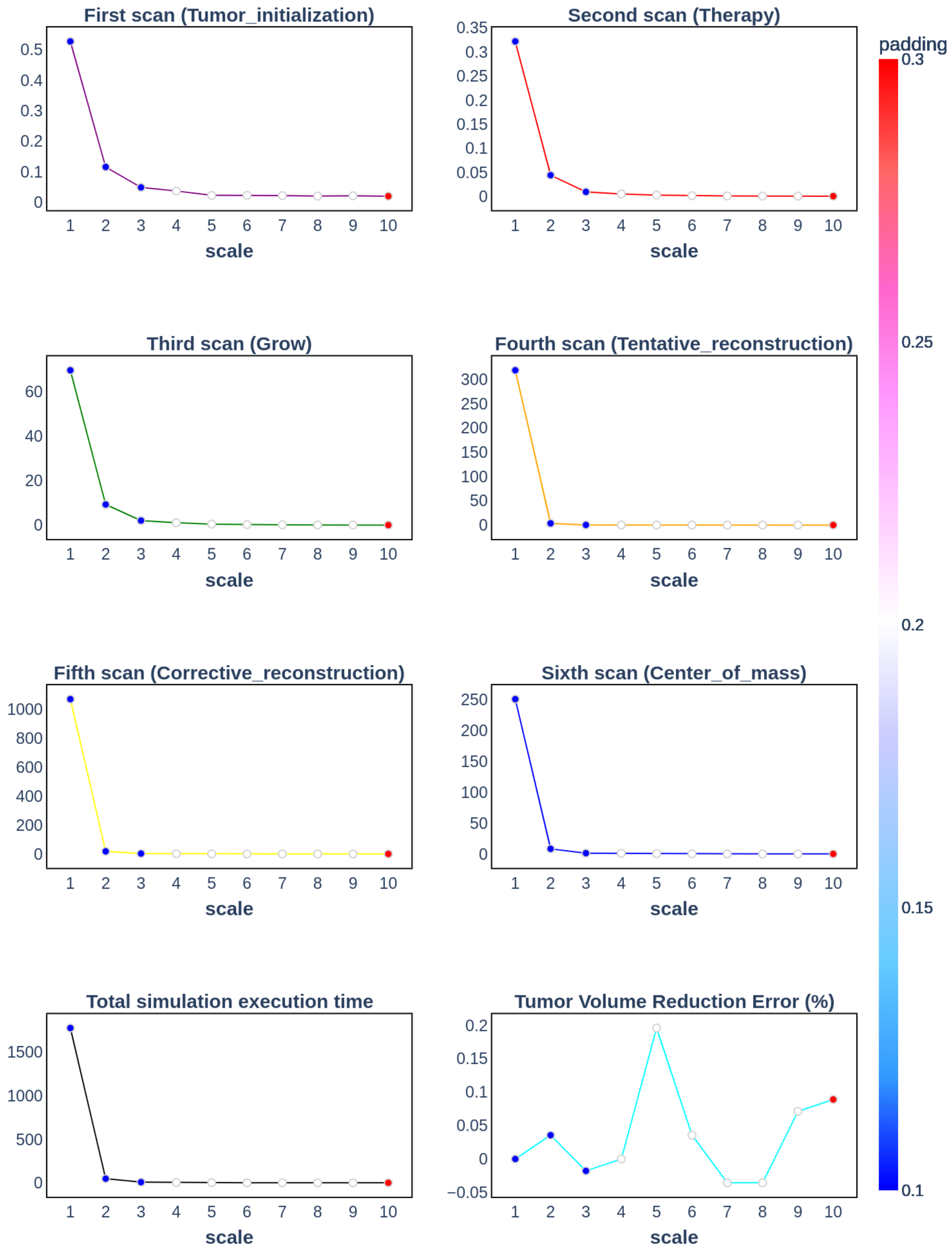


**Figure 4.1:** The scatter plots that express the relationship between the scale factor value, the simulation times and the tumor volume reduction error for Patient_15. The tumor volume reduction error and the simulation times, both the total execution time, as well as the times that are occupied by the distinct simulation scans, are represented in the y axis. The x axis represents the scale factor values. The corresponding padding factor values are represented by the color of the scatter points and according to the depicted colorscale.

| scale | padding | GC-dimension | x-dim | y-dim | z-dim | Total simulation execution time (*sec*) | Tumor Volume Reduction Error (*percentage*) | 3D Tumor Volume ($mm^3$) | Padded Space (*percentage*) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.1 | 1 | 106 | 114 | 118 | 8748.07 | 0 | 1425912 | 364.49 |
| 2 | 0.1 | 2 | 53 | 57 | 59 | 103.09 | 0.02 | 1425912 | 364.49 |
| 3 | 0.1 | 3 | 35 | 38 | 39 | 16.73 | 0.02 | 1400490 | 356.21 |
| 4 | 0.1 | 4 | 26 | 28 | 30 | 6.09 | 0.04 | 1397760 | 355.32 |
| 5 | 0.1 | 5 | 21 | 23 | 24 | 3.21 | 0.02 | 1449000 | 372.01 |
| 6 | 0.1 | 6 | 18 | 19 | 20 | 2 | 0.12 | 1477440 | 381.27 |
| 7 | 0.1 | 7 | 15 | 16 | 17 | 1.45 | 0.07 | 1399440 | 355.86 |
| 8 | 0.2 | 8 | 16 | 16 | 17 | 1.04 | 0 | 2228224 | 625.84 |
| 9 | 0.2 | 9 | 14 | 15 | 15 | 0.78 | -0.04 | 2296350 | 648.03 |
| 10 | 0.2 | 10 | 12 | 13 | 14 | 0.59 | 0.02 | 2184000 | 611.43 |

**Table 4.3:** The results of the resolution adaptation algorithm for Patient_16.
The recorded scale factor values range from 1 to 10. Each scale factor value corresponds to a padding factor value that ensures the normal simulation execution. The tumor volume reduction error fluctuates with the scale factor value increase, while the total simulation execution time decreases.

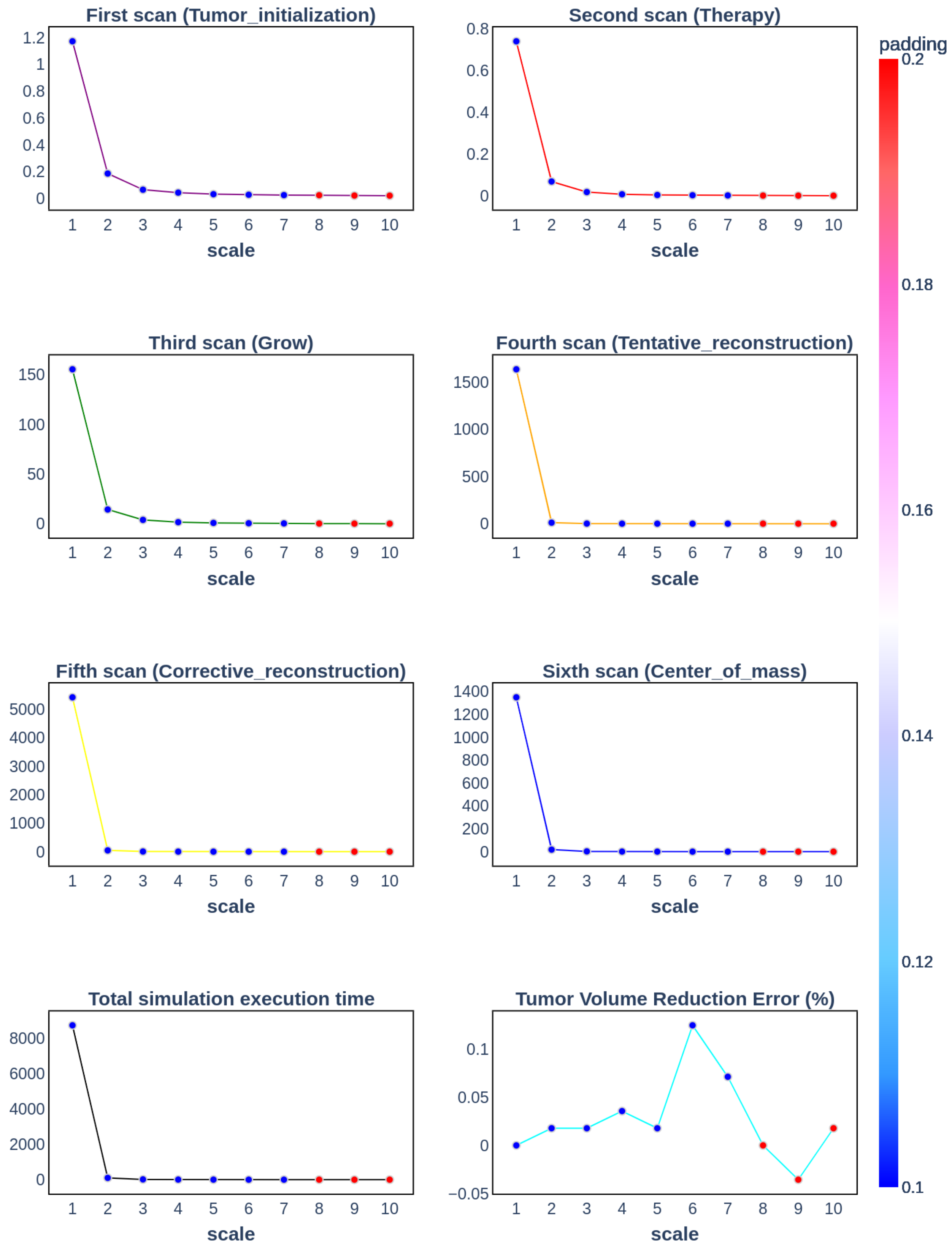


**Figure 4.2:** The scatter plots that express the relationship between the scale factor value, the simulation times and the tumor volume reduction error for Patient_16. The tumor volume reduction error and the simulation times, both the total execution time, as well as the times that are occupied by the distinct simulation scans, are represented in the y axis. The x axis represents the scale factor values. The corresponding padding factor values are represented by the color of the scatter points and according to the depicted colorscale.

| **scale** | **padding** | **GC-dimension** | **x-dim** | **y-dim** | **z-dim** | **Total simulation execution time** (*sec*) | **Tumor Volume Reduction Error** (*percentage*) | **3D Tumor Volume** ($mm^3$) | **Padded Space** (*percentage*) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 118 | 123 | 132 | 45440.28 | 0 | 1915848 | 256.95 |
| 2 | 0 | 2 | 59 | 62 | 66 | 492.36 | -0.04 | 1931424 | 259.85 |
| 3 | 0.1 | 3 | 47 | 50 | 53 | 66.03 | -0.02 | 3362850 | 526.54 |
| 4 | 0.1 | 4 | 36 | 37 | 40 | 16.22 | 0.02 | 3409920 | 535.31 |
| 5 | 0.1 | 5 | 28 | 30 | 32 | 7.27 | -0.02 | 3360000 | 526.01 |
| 6 | 0.1 | 6 | 24 | 25 | 27 | 4.24 | 0.05 | 3499200 | 551.94 |
| 7 | 0.1 | 7 | 20 | 21 | 23 | 2.69 | 0.02 | 3313380 | 517.32 |
| 8 | 0.1 | 8 | 18 | 19 | 20 | 2.14 | 0.02 | 3502080 | 552.48 |
| 9 | 0.1 | 9 | 16 | 17 | 18 | 1.32 | -0.05 | 3569184 | 564.98 |
| 10 | 0.1 | 10 | 14 | 15 | 16 | 1.11 | 0.07 | 3360000 | 526.01 |

**Table 4.4:** The results of the resolution adaptation algorithm for Patient_17.
The recorded scale factor values range from 1 to 10. Each scale factor value corresponds to a padding factor value that ensures the normal simulation execution. The tumor volume reduction error fluctuates with the scale factor value increase, while the total simulation execution time decreases.

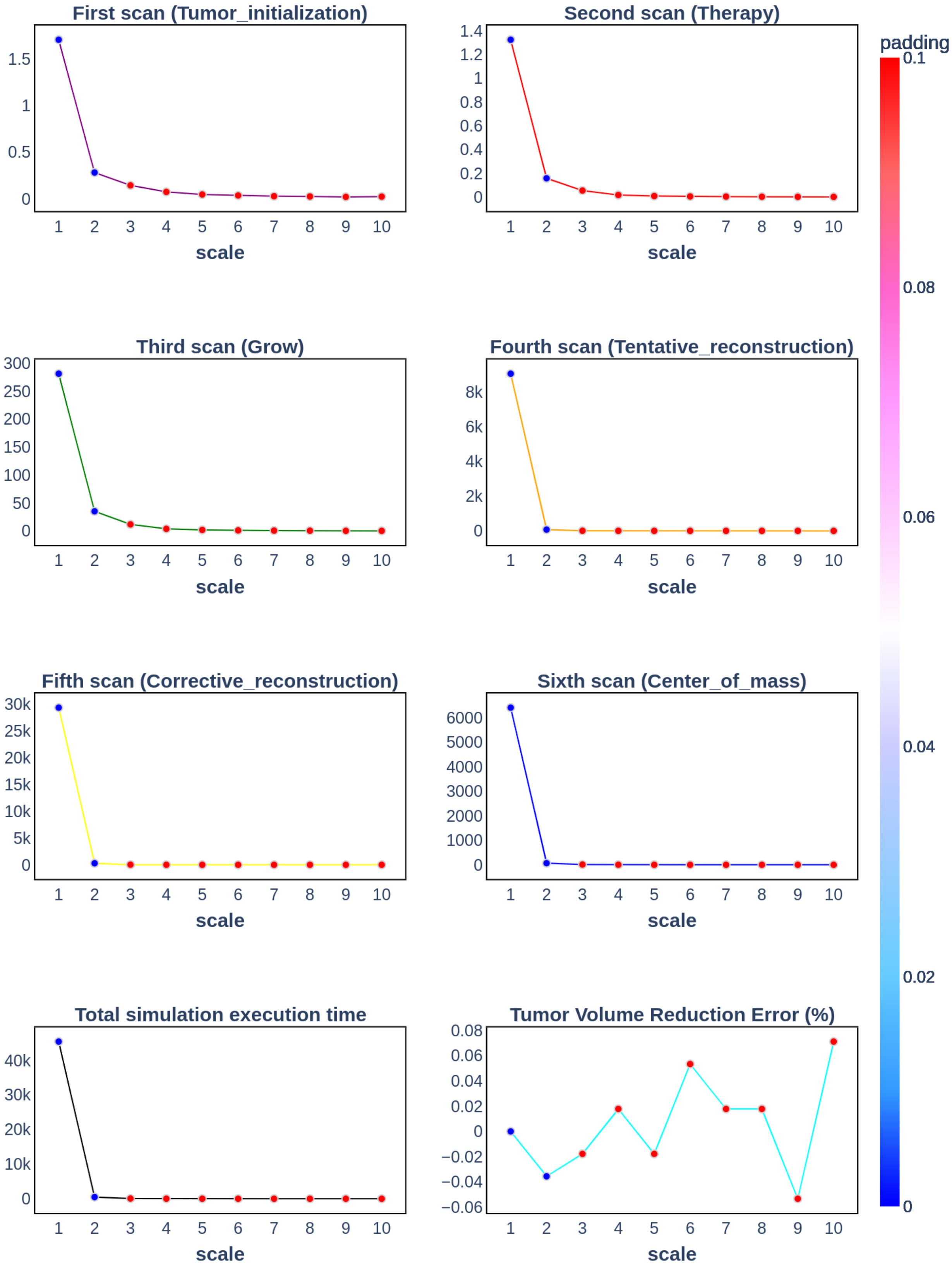


**Figure 4.3:** The scatter plots that express the relationship between the scale factor value, the simulation times and the tumor volume reduction error for Patient_17. The tumor volume reduction error and the simulation times, both the total execution time, as well as the times that are occupied by the distinct simulation scans, are represented in the y axis. The x axis represents the scale factor values. The corresponding padding factor values are represented by the color of the scatter points and according to the depicted colorscale.

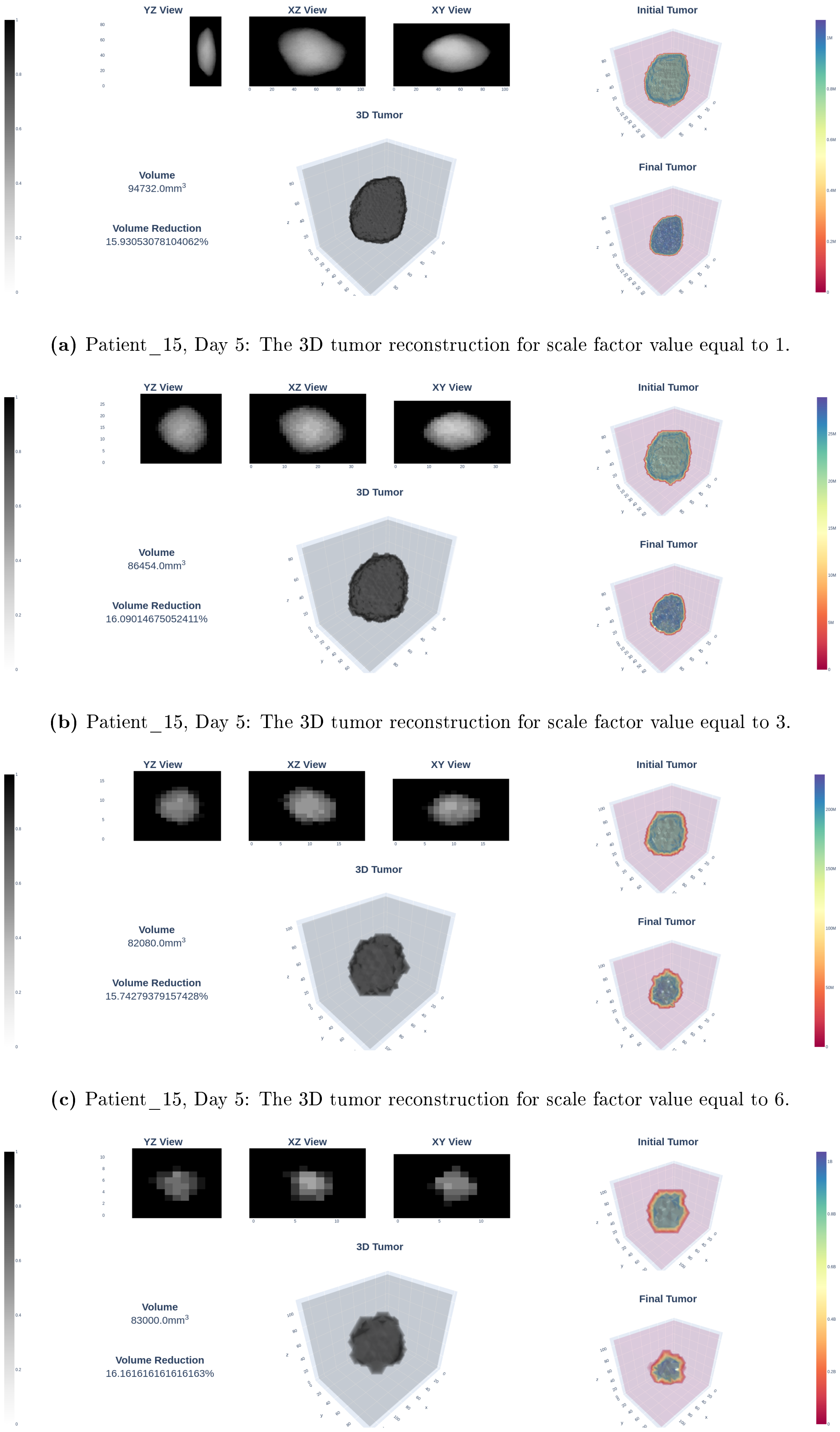
### 4.1.3 A comparison of the simulation evolution for the varying resolution versions that correspond to the distinct scale and padding factor sets for each dataset patient

The 3D reconstruction of the simulated tumor, along with the corresponding 2D plane views, the initial tumor reconstruction and the final tumor reconstruction, is displayed for the execution of the simulation for each dataset patient, with four distinct values for the input file scale factor ($scale = 1$, $scale = 3$, $scale = 6$ and $scale = 10$) and for two distinct time points of the simualtion (day 5 and day 20), in **Figures 4.4-4.9**.

Reviewing the 2D plane views for each dataset patient and for a specific simulation time point (day 5 and day 20 respectively), the effects of the increase of the scale factor of the geometrical cell size are evident. As the scale factor increases, the size of the input (i.e. the size of the 3D matrix along each axis) decreases and each geometrical cell spans a bigger region. This means that a distinct value (0xFF for tumor tissue and 0x00 for normal tissue) is assigned to a bigger region, which causes the image resolution to decrease. Moreover, the increase of the scale factor causes a potential increase in the padding factor, which is also observed in the 2D plane views, especially for padding along the x axis (YZ view).

The image resolution is indicative of the effects of the scale factor to the simulation execution. However, it should not be confused for the simulation resolution, which aside from the size and shape of the tumor also refers to the histology of the tumor. In that respect, the simulation resolution encompasses the image resolution and the scale factor values that correspond to bad image resolution are safely assumed to correspond to inadequate simulation resolution too.

Reviewing **Figures 4.4-4.9**, the values for the scale factor that are greater than 3 are rejected as unsuitable, since they correspond to bad image resolution and subsequently inadequate simulation resolution. That fact is not mirrored in the recorded tumor volume reduciton percentage that is used as the criterion of the resolution adaptation algorithm and stays mainly unaffected by the increase of the scale factor value, when it is limited to small values (here up to 10). This means that the reconstructions of **Figures 4.4-4.9** should also be taken into account, along with the error of **Figures 4.1-4.3**, when the optimal scale factor for each dataset patient is chosen.



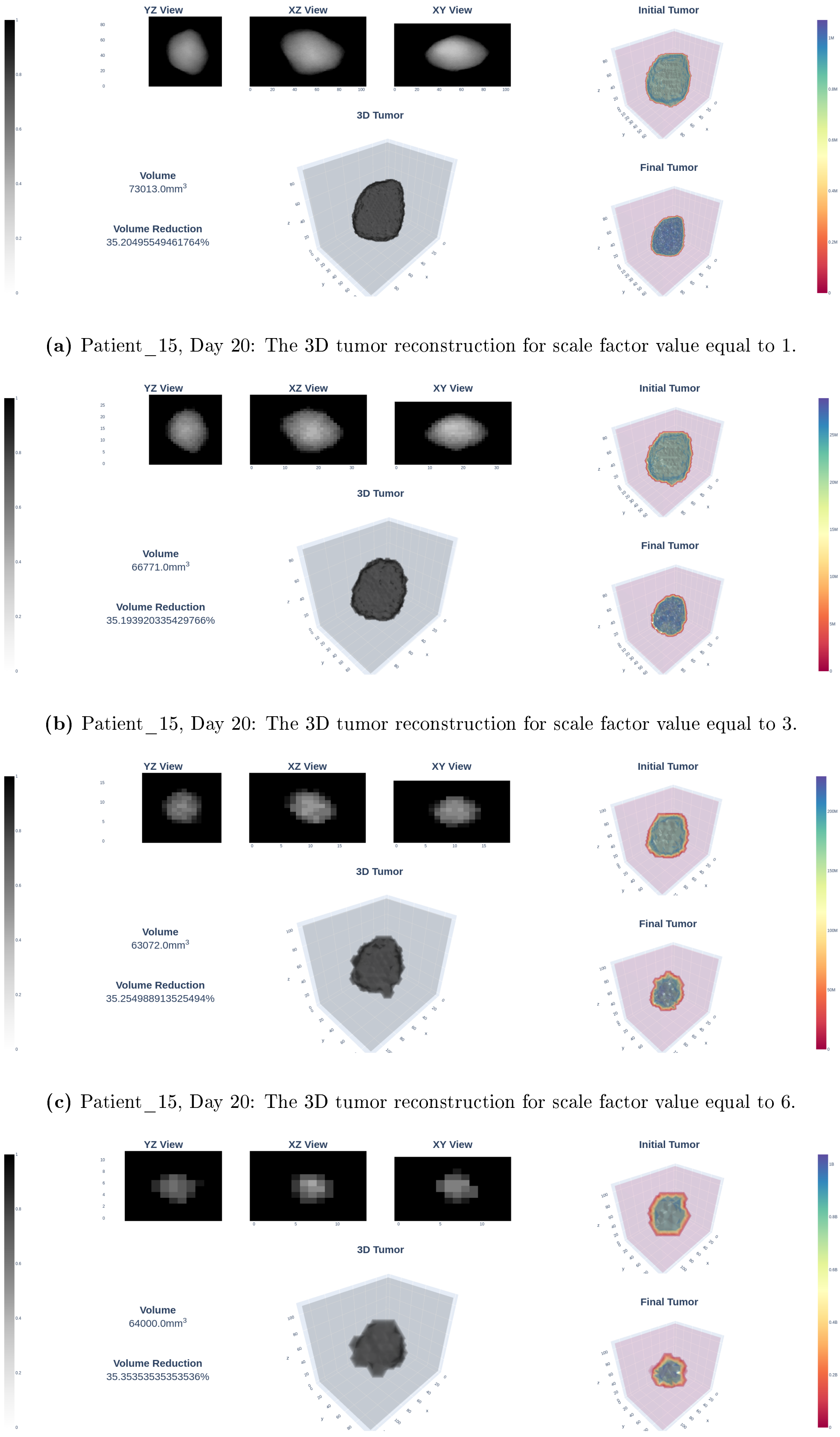
**(a)** Patient_15, Day 5: The 3D tumor reconstruction for scale factor value equal to 1.

**(b)** Patient_15, Day 5: The 3D tumor reconstruction for scale factor value equal to 3.

**(c)** Patient_15, Day 5: The 3D tumor reconstruction for scale factor value equal to 6.

**(d)** Patient_15, Day 5: The 3D tumor reconstruction for scale factor value equal to 10.

**Figure 4.4:** The 3D tumor reconstruction for varying scale factor values at day 5 of the simulation for Patient_15.



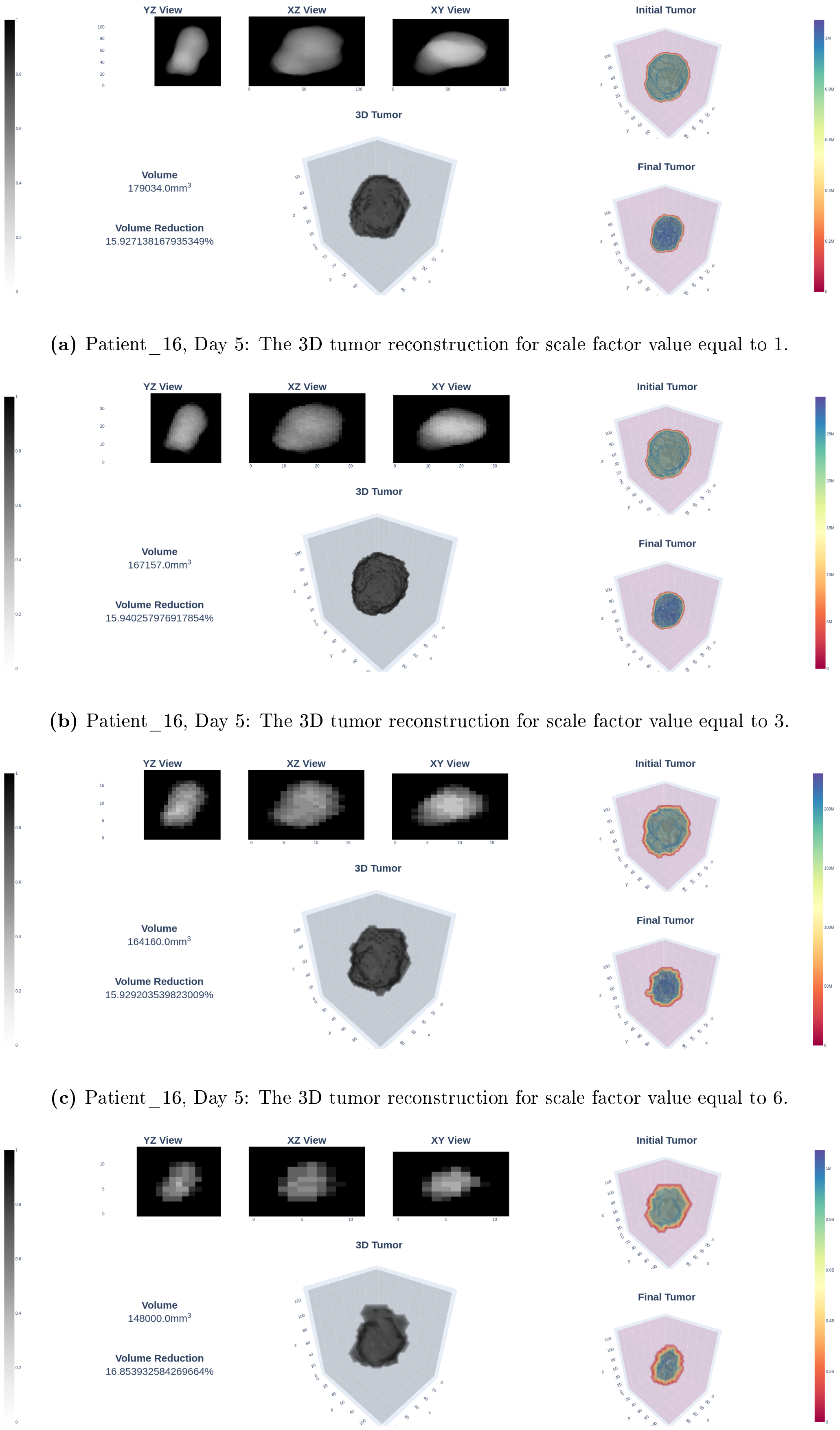
**(a)** Patient_15, Day 20: The 3D tumor reconstruction for scale factor value equal to 1.

**(b)** Patient_15, Day 20: The 3D tumor reconstruction for scale factor value equal to 3.

**(c)** Patient_15, Day 20: The 3D tumor reconstruction for scale factor value equal to 6.

**(d)** Patient_15, Day 20: The 3D tumor reconstruction for scale factor value equal to 10.

**Figure 4.5:** The 3D tumor reconstruction for varying scale factor values at day 20 of the simulation for Patient_15.



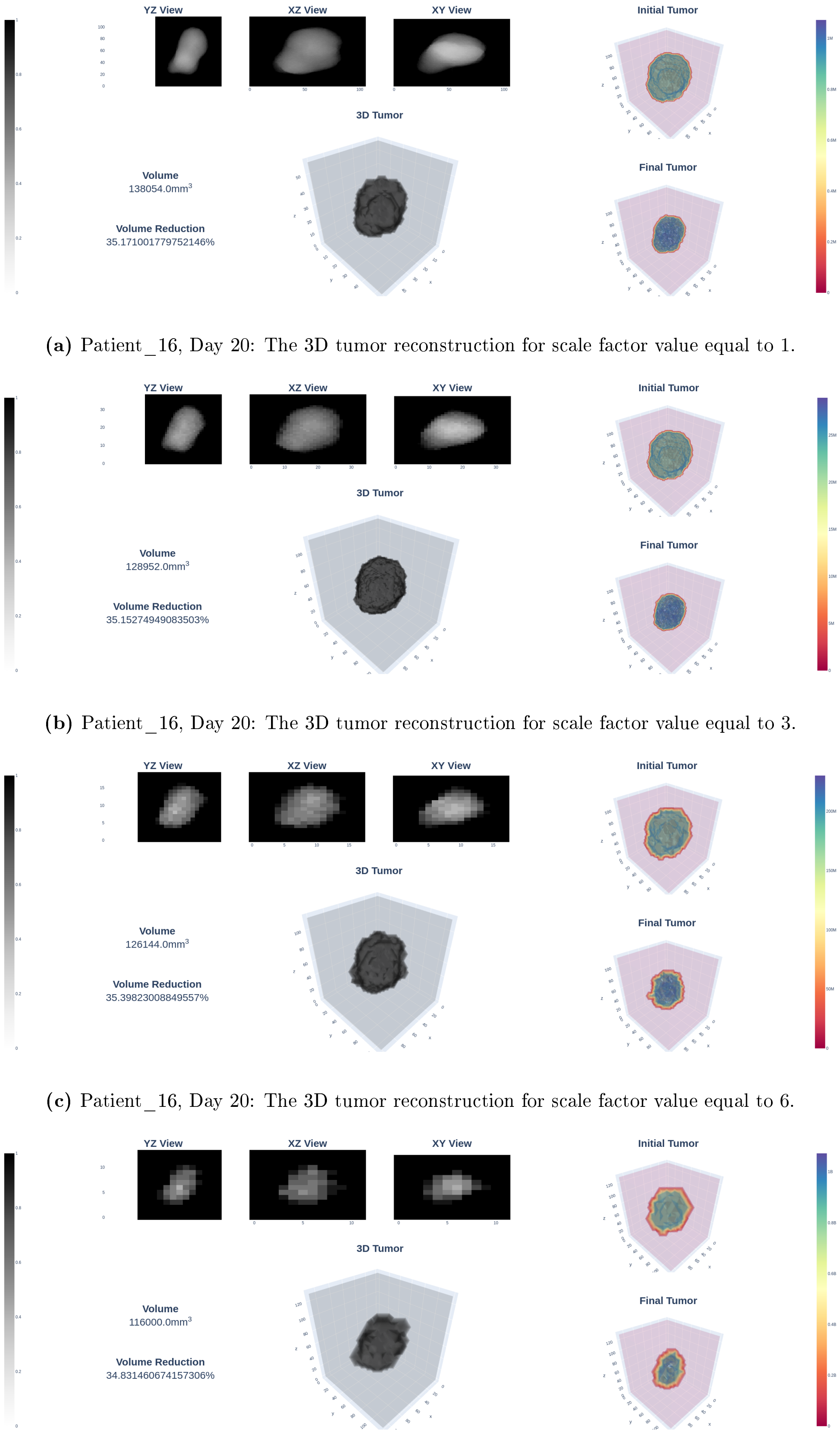
**(a)** Patient_16, Day 5: The 3D tumor reconstruction for scale factor value equal to 1.

**(b)** Patient_16, Day 5: The 3D tumor reconstruction for scale factor value equal to 3.

**(c)** Patient_16, Day 5: The 3D tumor reconstruction for scale factor value equal to 6.

**(d)** Patient_16, Day 5: The 3D tumor reconstruction for scale factor value equal to 10.

**Figure 4.6:** The 3D tumor reconstruction for varying scale factor values at day 5 of the simulation for Patient_16.



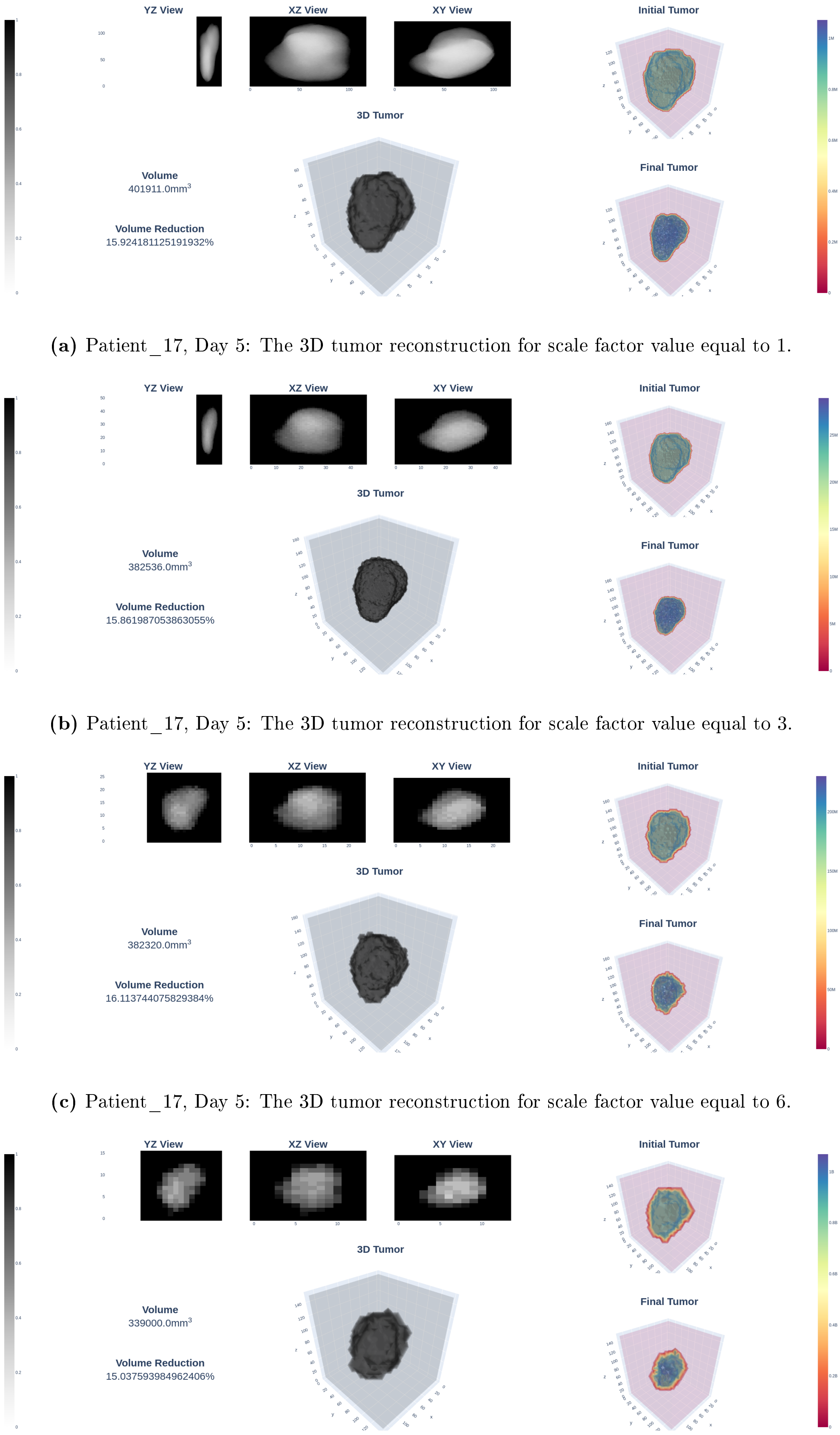
**(a)** Patient_16, Day 20: The 3D tumor reconstruction for scale factor value equal to 1.

**(b)** Patient_16, Day 20: The 3D tumor reconstruction for scale factor value equal to 3.

**(c)** Patient_16, Day 20: The 3D tumor reconstruction for scale factor value equal to 6.

**(d)** Patient_16, Day 20: The 3D tumor reconstruction for scale factor value equal to 10.

**Figure 4.7:** The 3D tumor reconstruction for varying scale factor values at day 20 of the simulation for Patient_16.



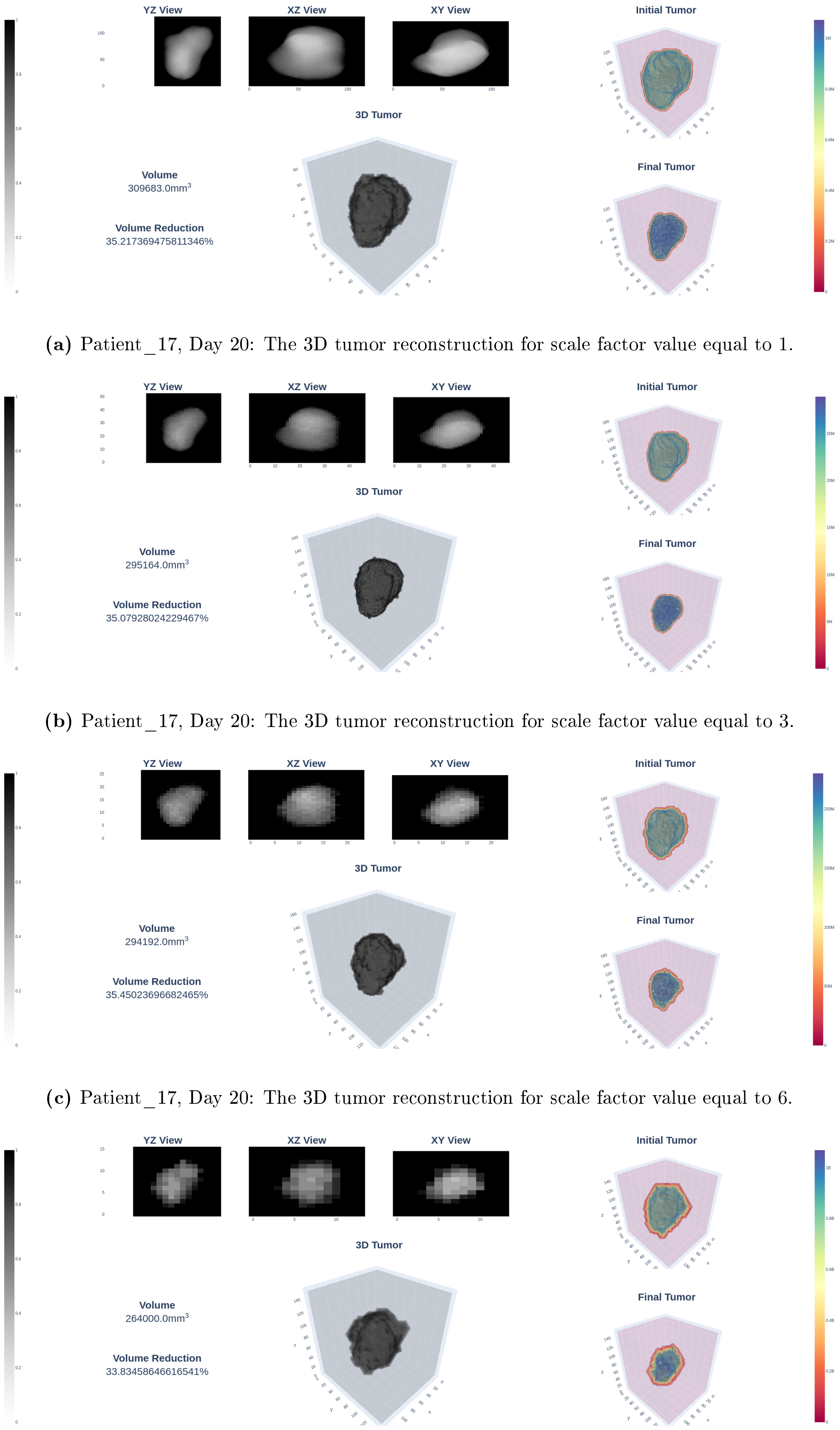
**(a)** Patient_17, Day 5: The 3D tumor reconstruction for scale factor value equal to 1.

**(b)** Patient_17, Day 5: The 3D tumor reconstruction for scale factor value equal to 3.

**(c)** Patient_17, Day 5: The 3D tumor reconstruction for scale factor value equal to 6.

**(d)** Patient_17, Day 5: The 3D tumor reconstruction for scale factor value equal to 10.

**Figure 4.8:** The 3D tumor reconstruction for varying scale factor values at day 5 of the simulation for Patient_17.



**(a)** Patient_17, Day 20: The 3D tumor reconstruction for scale factor value equal to 1.

**(b)** Patient_17, Day 20: The 3D tumor reconstruction for scale factor value equal to 3.

**(c)** Patient_17, Day 20: The 3D tumor reconstruction for scale factor value equal to 6.

**(d)** Patient_17, Day 20: The 3D tumor reconstruction for scale factor value equal to 10.

**Figure 4.9:** The 3D tumor reconstruction for varying scale factor values at day 20 of the simulation for Patient_17.

### 4.1.4 The chosen optimal resolution adaptation and the corresponding value for the scale and padding factor set for each dataset patient

Reviewing the results of the execution of the simulation resolution adaptation algorithm of **Figure 3.3** for each dataset patient, the scale factor and padding factor values that optimally address the tradeoff between the simulation resolution and the simulation cost (execution time) were chosen as shown in **Table 4.5**.

The set of values for the scale and padding factors was chosen for each dataset patient according to the following criteria:

- The tumor volume reduciton error that is introduced by the decrease of the simulation resolution is not greater than the algorithm specified tolerable error (10%). Reviewing **Figures 4.1-4.3**, it becomes evident that all the recorded scale factor values satisfy that standard.
- The image resolution is not greatly affected. Reviewing **Figures 4.4-4.9**, the values for the scale factor that are greater than 3 are rejected.
- The simulation execution time is not greater than a few minutes.

| **Pseudonym** | Patient_15 | Patient_16 | Patient_17 |
|---|---|---|---|
| **scale** | 2 | 2 | 3 |
| **padding** | 0.1 | 0.1 | 0.1 |
| **GC-dimension** | 2 | 2 | 3 |
| **x-dim** | 52 | 53 | 47 |
| **y-dim** | 41 | 57 | 50 |
| **z-dim** | 46 | 59 | 53 |
| **Tumor Volume Reduction Error** (*percentage*) | 0.04 | 0.02 | -0.02 |
| **Total simulation execution time** (*sec*) | 47.77 | 103.09 | 66.03 |

**Table 4.5:** The chosen set of values for the scale factor and the padding factor per dataset patient. These values define the input files that are used for the simulation of the patients in the following steps of the workflow. They were chosen to optimally address the tradeoff of the simulation resolution and the simualtion cost. The tolerable error that the decrease in the simulation resolution introduces is limited to 10%. The image resolution is also evaluated, which liimits the greatest allowed value for the scale factor to 3. The target maximum execution time for a single simulation is a few minutes.

The set of values for the scale factor and the padding factor, as shown in **Table 4.5**, define unambiguously the input files that are used for the simulation of the dataset patients in the following steps of the workflow.

## 4.2 Sensitivity Analysis

### 4.2.1 The evaluated with respect to the initial tumor composition results of the graphical method for sensitivity analysis

The sensitivity analysis process was performed for all the versions of the source code for the Nephroblastoma Oncosimulator, using the graphical method that was presented in detail in the dedicated chapter of **Methodology**. The results presented here refer to the analysis that was performed for the performance enhanced version of the Nephroblastoma Oncosimulator, i.e. the version for which at each simulation time step the computations for the third scan of the algorithm are performed concurrently for the geometrical cells of the 3D matrix that represents the solid tumor in silico. Moreover, the presented results refer to Patient_15 and the corresponding input files that optimally address the simulation resolution and costs tradeoff, as they were determined in the previous data preprocessing step.

The results of the graphical method for sensitivity analysis that was performed for the performance enhanced version of the Nephroblastoma Oncosimulator are displayed in **Figures 4.10** with respect to the initial tumor composition, as measured by the percentage of each cell category, proliferative, dormant, differentiated and dead, inside the initial tumor (see **Table 3.1**).

The scatter plots displayed in **Figure 4.10** express the dependence of the initial composition of the tumor, before the effects of therapy, from the values of the various simulation input parameters. This is achieved by plotting the initial cell categories population percentages against the values for the various simulation input parameters. These plots also offer an insight on the effect of the simulation input parameters on the free growth of the tumor, since the cell population percentages of the different cell categories are stabilized inside the tumor after a relatively short time when no external stimulus (e.g. therapy) is applied.

The plots shown in **Figure 4.10** are compliant with the ones produced in previous work for the predecessor verified model (see **Figure 3.5**). A brief examination of the plots in **Figure 4.10** highlights the effect of some key parameters to the initial composition of the tumor and leads to intuitively logical conclusions. More specifically:

- Increase in the value of the cell cycle duration ($T_c$) causes increase in the initial population percentage of the proliferative cells of the tumor and decrease in the initial population percentage of the differentiated cells of the tumor. This is an expected outcome, since in the cases where the cell cycle lasts longer, for the same amount of time more STEM cells are concentrated in the $G_1$, $S$, $G_2$ and $M$ phases and less complete the mitosis that could produce LIMP cells with assymetric division and in the long run more differentiated cells.

- Increase in the value of the maximum duration of the $G_0$ phase ($T_{G0}$) causes increase in the initial population percentage of the dormant cells of the tumor and decrease in the initial population percentage of the differentiated cells of the tumor. This is also an expected outcome, since the longer the time that STEM cells spend in the $G0$ phase the more the concentration of the cells in this phase (population percentage of the dormant cells) increases. At the same time, an indirect impact of the increase of the $T_{G0}$ time is the decrease of differentiated cells in the tumor, since when more cells are concentrated in the $G0$ phase, less cells are going through the mitosis process that could in the long run produce LIMP and differentiated cells.

- Increase in the values of the time duration for the completion of apoptosis and necrosis ($T_N$ and $T_A$ respectively) causes increase in the initial population percentage of the dead cells of the tumor and decrease in the initial population percentage of the differentiated cells of the tumor. This is an intuitively logical outcome, since the $T_N$ and $T_A$ times practically express the time needed for the products of necrosis and apoptosis respectively to exit the tumor.

- Increase in the values of the necrosis and apoptosis rates of the differentiated cells ($R_{NDiff}$ and $R_{ADiff}$ respectively) causes decrease in the initial population percentage of the differentiated cells of the tumor, which is accompanied by increase in the initial population percentage of the rest cell categories. This is once again an intuitive outcome, since $R_{NDiff}$ and $R_{ADiff}$ express the speed in which the differentiated cells are subject to necrosis and apoptosis respectively, which indicates that higher values for these parameters cause the differentiated cells to die more frequently and as a result their population percentage to decrease and the dead population percentage to increase, as well as the population percentage for the remaining two cell categories to also increase, since there is more free space for them.

- Increase in the value of the percentage of symmetric divisions ($P_{sym}$) causes increase in the initial population percentage of the proliferative cells of the tumor and decrease in the initial population percentage of the differentiated cells of the tumor. This is due to the increase of the STEM cells and the dicrease of the LIMP cells that the increase of the percentage of symmetric divisions causes, since proliferative cells are a subset of the STEM cell category and differentiated cells are the main product of the LIMP cell category. This means that higher values for $P_{sym}$ correspond to more aggresive tumors, which is also intuitively inferred.

- Increase in the value of the percentage of the STEM cells that enter the $G_0$ phase ($P_{sleep}$) causes increase in the initial population percentage of the differentiated cells of the tumor and decrease in the initial population percentage of the proliferative cells of the tumor. This is the expected outcome, since $P_{sleep}$ expresses the probability with which STEM cells enter the dormant $G0$ phase and as a result increase in the value of $P_{sleep}$ causes more STEM cells to enter the $G0$ phase and an increase in the concentration of the cells that reside in it. This also means that higher values for $P_{sleep}$ correspond to less aggresive tumors, which is also intuitively inferred.

- The rest of the simulation input parameters do not have a considerable impact on the initial composition of the tumor.

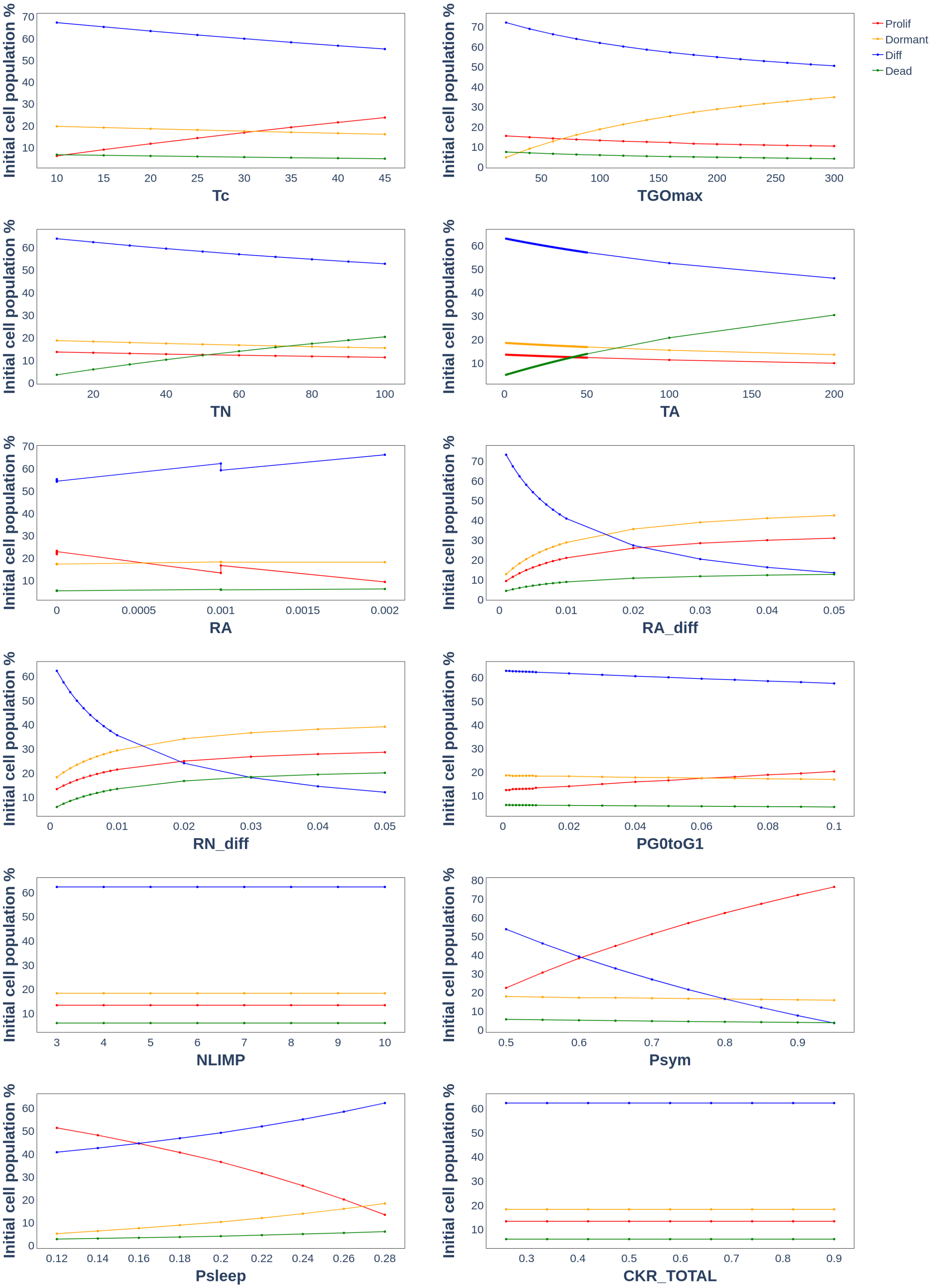


**Figure 4.10:** The graphical method sensitivity analysis results for the optimized version of the Nephroblastoma Oncosimulator and with respect to the initial tumor composition. The simulations were executed with the resolution adapted for Patient_15 and with the computations for the third scan of the algorithm performed concurrently for the geometrical cells. The results are compliant with the ones for the verified predecessor model (**Figure 3.5**), which allows the safe verification of the current model.

### 4.2.2 The evaluated with respect to the tumor volume variation results of the graphical method for sensitivity analysis

The sensitivity analysis process was performed for all the versions of the source code for the Nephroblastoma Oncosimulator, using the graphical method that was presented in detail in the dedicated chapter of **Methodology**. The results presented here refer to the analysis that was performed for the performance enhanced version of the Nephroblastoma Oncosimulator, i.e. the version for which at each simulation time step the computations for the third scan of the algorithm are performed concurrently for the geometrical cells of the 3D matrix that represents the solid tumor in silico. Moreover, the presented results refer to Patient_15 and the corresponding input files that optimally address the simulation resolution and costs tradeoff, as they were determined in the previous data preprocessing step.

The results of the graphical method for sensitivity analysis that was performed for the performance enhanced version of the Nephroblastoma Oncosimulator are displayed in **Figures 4.11** with respect to the tumor volume variation, as measured by the relative tumor volume variation percentage (see **Table 3.1** and **Equation 2.9**).

The scatter plots displayed in **Figure 4.11** express the dependence of the volume reduction of the tumor, after the effects of therapy and the completion of the simulation, from the values of the various simulation input parameters. This is achieved by plotting the relative volume variation of the tumor compared to its volume at the start of the simulation against the values for the various simulation input parameters.

The plots shown in **Figure 4.11** are compliant with the ones produced in previous work for the verified predecessor model only for a subset of the simulation input parameters. More specifically, $T_N$, $T_A$, $N_{LIMP}$, $R_{ADiff}$, $R_{NDiff}$ and $CKR_{TOTAL}$ have an effect in the volume reduction of the tumor that is similar to the one the predecessor model has (see **Figure 3.4**). The change of the values of the first five parameters does not affect the therapy induced volume variation percentage of the tumor, which retains a constant value of $-60\%$ approximately. The $CKR_{TOTAL}$ parameter is the main parameter that greatly affects the outcome of the simulation in the means of tumor volume reduction. As the value for $CKR_{TOTAL}$ increases the relative variation percentage of the tumor decreases in the negative, which is interpreted as an increase of the volume reduction percentage. This observation is also intuitively inferred, since increase of the cell kill ratio results in more tumor cells to be targeted, hit and exterminated by therapy.

The rest of the simulation input parameters behave in the same manner as $T_N$, $T_A$, $N_{LIMP}$, $R_{ADiff}$, $R_{NDiff}$, with the relative variation percentage of the volume remaining consant at $-60\%$ approximately for their whole range of values. However, increase in the values of $T_{G0}$, $P_{G0toG1}$ and $P_{sym}$ has been observed, in previous sensitivity analysis with the graphical method, to cause increase in the value of the relative volume variation percentage of the tumor, while increase in the values of $T_c$, $R_A$ and $P_{sleep}$ has been observed to cause decrease in the value of the relative volume variation percentage of the tumor.

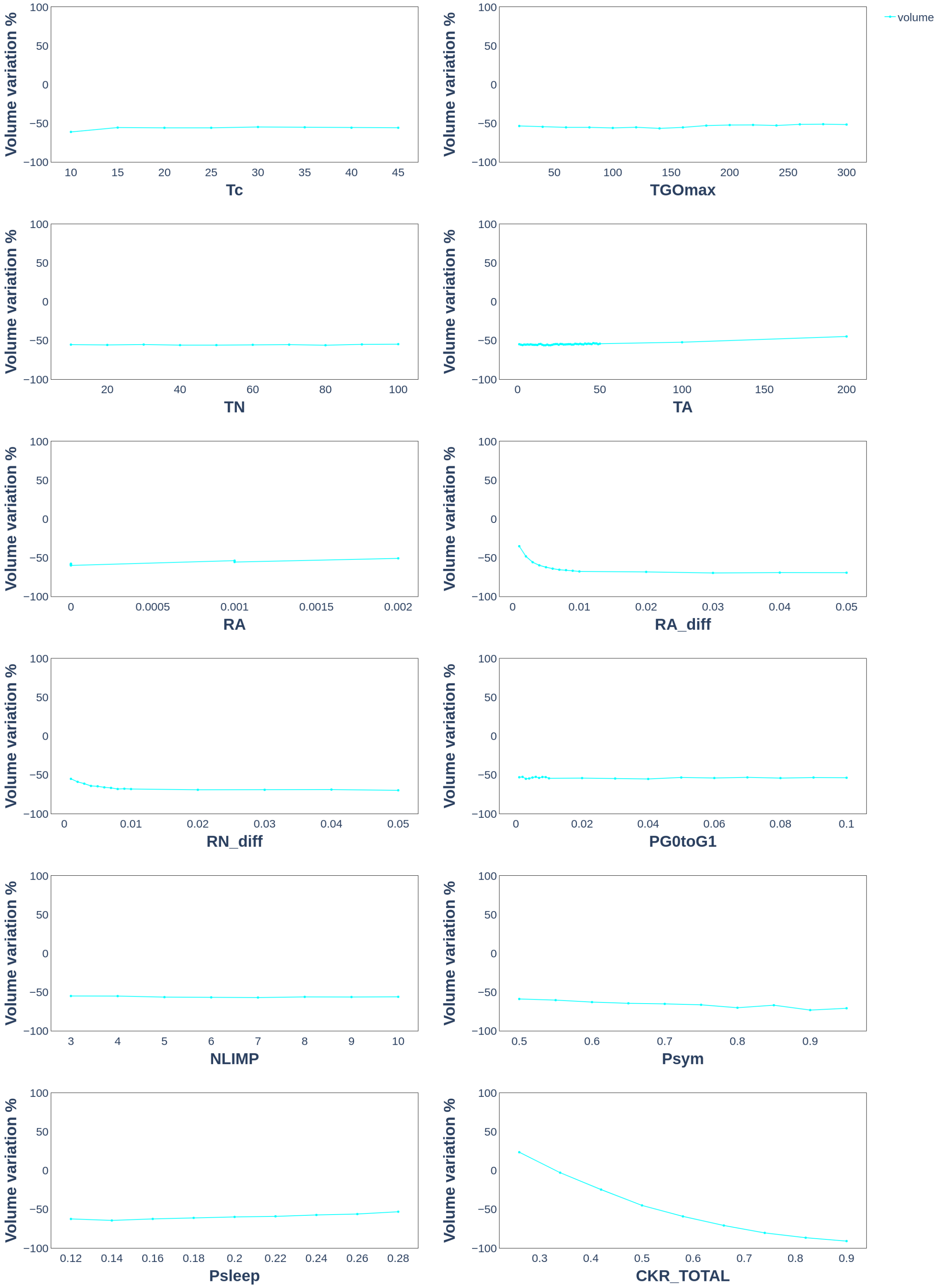


**Figure 4.11:** The graphical method sensitivity analysis results for the optimized version of the Nephroblastoma Oncosimulator and with respect to the tumor volume variation. The simulations were executed with the resolution adapted for Patient_15 and with the computations for the third scan of the algorithm performed concurrently for the geometrical cells. The results are compliant with the ones for the verified predecessor model (**Figure 3.4**), which allows the safe verification of the current model.

However, this divergence from the typical behavior for some of the input parameters with concern to the manner in which their values affect the volume reduction of the tumor should not be immediately interpreted as a problem in the functionality of the current version of the model. The sensitivity analysis that was performed for the current model does not lead to counterintuitive observations and conclusions. The one parameter that intuitively has the greatest impact on the volume reduction of the tumor, $CKR_{TOTAL}$, retains the behavior observed in the previous sensitivity analysis performed with the graphical method. The rest of the parameters seem to have no impact on the volume reduction of the tumor, which is intuitively logical, since they do not have a role in the therapeutic scheme. There exists the possibility that their effect has been overshadowed by the impact of $CKR_{TOTAL}$.

### 4.2.3 The evaluated with respect to the tumor volume variation results of the graphical method for sensitivity analysis

The sensitivity analysis process was performed for all the versions of the source code for the Nephroblastoma Oncosimulator, using the graphical method that was presented in detail in the dedicated chapter of **Methodology**. The results presented here refer to the analysis that was performed for the performance enhanced version of the Nephroblastoma Oncosimulator, i.e. the version for which at each simulation time step the computations for the third scan of the algorithm are performed concurrently for the geometrical cells of the 3D matrix that represents the solid tumor in silico. Moreover, the presented results refer to Patient_15 and the corresponding input files that optimally address the simulation resolution and costs tradeoff, as they were determined in the previous data preprocessing step.

The results of the graphical method for sensitivity analysis that was performed for the performance enhanced version of the Nephroblastoma Oncosimulator are displayed in **Figures 4.12** with respect to the final tumor composition, as measured by the relative population variation percentages for each cell category, proliferative, dormant, differentiated and dead, in the final tumor (see **Table 3.1** and **Equation 3.3**).

The scatter plots displayed in **Figure 4.12** express the dependence of the final composition of the tumor, after the effects of therapy and the completion of the simulation, from the values of the various simulation input parameters. This is achieved by plotting the therapy induced relative variation percentage for each cell category against the values for the various simulation input parameters.

The plots shown in **Figure 4.12** are overall compliant with the ones produced for the verified predecessor model (see **Figure 3.6**). More specifically, the behavior of the simulation for the range of values of the $T_N$, $T_A$, $R_A$, $N_{LIMP}$ and $CKR_{TOTAL}$ parameters is identical to the one observed for the verified predecessor model. The behavior for the range of values of the $T_c$, $T_{G0}$, $R_{ADiff}$, $R_{NDiff}$, $P_{sym}$ and $P_{sleep}$ is also similar, in the means that the variation of the values for each one of these parameters affects the population variation percentage of the different cell categories in the manner that has been previously observed for the predecessor model. The difference lies in the absolute values for the percentages for the various cell categories, especially concerning the category of the dead cells, whose population variation percentage has lower values (its plot is moved downwards on the vertical axis by a constant) than the ones observed in the past. This agrees with the unexpected behavior for the volume reduction of the tumor that was observed in the current sensitivity analysis for these parameters. That fact highlights the consistent manner in which the current model behaves.

A brief examination of the plots in **Figure 4.12** highlights the effect of some key parameters to the final composition of the tumor and leads to intuitively logical conclusions. More specifically:

- Increase in the value of the maximum duration of the $G_0$ phase ($T_{G0}$) causes increase in the final population percentage of the dormant and dead cells of the tumor. This is due to the fact that increase in the $T_{G0}$ time causes more STEM cells to be concentrated in this phase, which is only hit by therapy with Vinctristine.

- Increase in the values of the time duration for the completion of apoptosis and necrosis ($T_N$ and $T_A$ respectively) causes increase in the final population percentage of the dead cells of the tumor. This is an intuitively logical outcome, since the $T_N$ and $T_A$ times practically express the time needed for the products of necrosis and apoptosis respectively to exit the tumor.

- Increase in the value of the percentage of symmetric divisions ($P_{sym}$) causes increase in the final population percentage of all the cell categories. This is an intuitive outcome, since higher values for $P_{sym}$ correspond to more aggresive tumors with cells that keep multiplying with time.

- Increase in the value of the percentage of the STEM cells that enter the $G_0$ phase ($P_{sleep}$) causes increase in the final population percentage of all the cell categories, except for the category of the dead cells, which decreases. This is due to the fact that higher values for $P_{sleep}$ lead to higher concentration of cells in the $G_0$, which can die through necrosis or apoptosis. This also is an intuitive outcome, since $P_{sleep}$ correspond to less aggresive tumors., which is also intuitively inferred.

- Increase in the value of the total cell kill ratio of the therapy ($CKR_{TOTAL}$) causes decrease in the final population percentage of all the cell categories, except for the category of the differentiated cells, which increases. This is an intuitively logical outcome, since the cell kill ratio expresses the percentage of cells that the therapy targets, hits and exterminates and as a result, higher values for the $CKR_{TOTAL}$ parameter causes the cells of the tumor to decrease in total. The differentiated cell category is the only one that is not affected, since it is not targeted by the therapy.

- The rest of the simulation input parameters do not have a considerable impact on the final composition of the tumor.

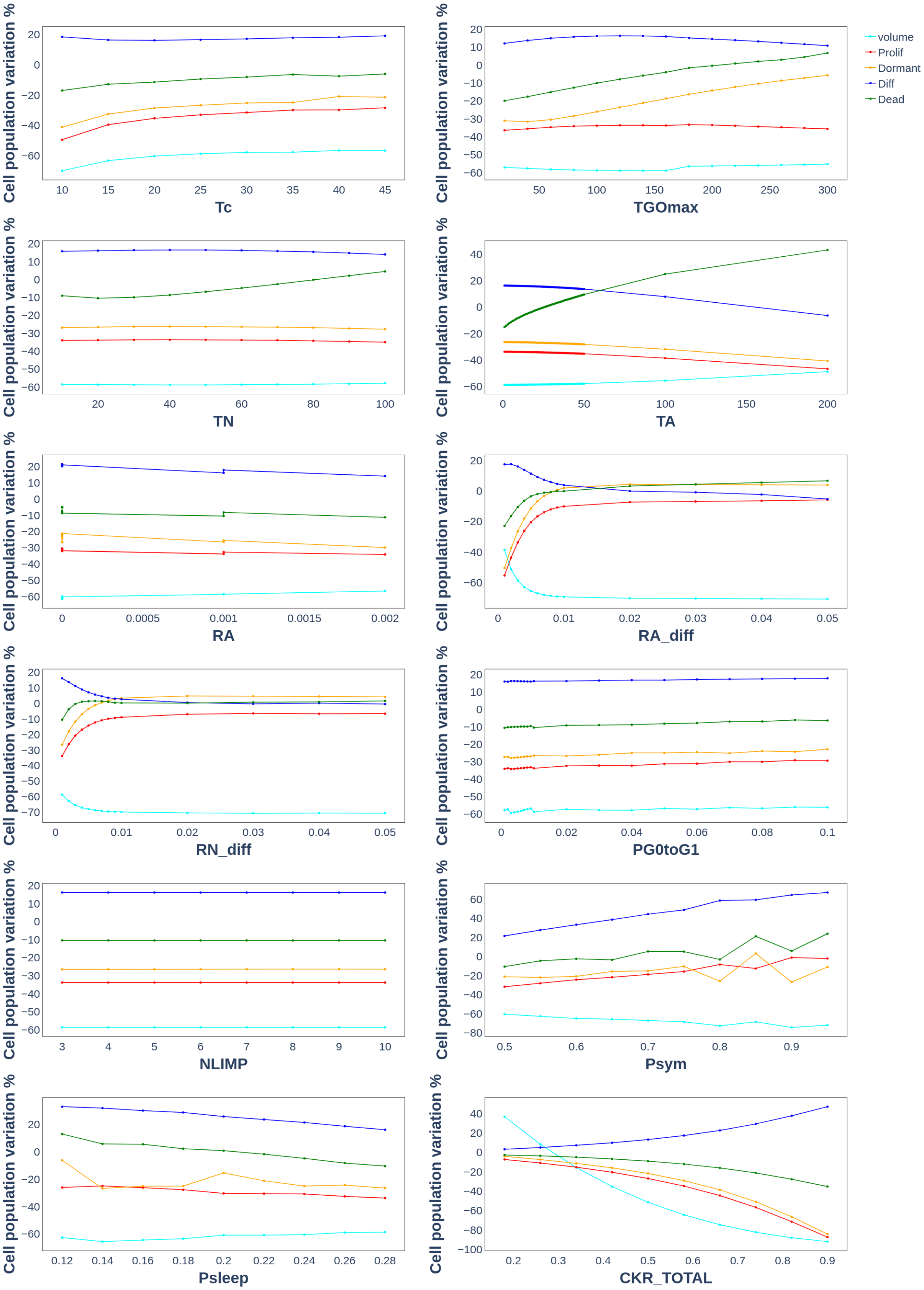


**Figure 4.12:** The graphical method sensitivity analysis results for the optimized version of the Nephroblastoma Oncosimulator and with respect to the final tumor composition. The simulations were executed with the resolution adapted for Patient_15 and with the computations for the third scan of the algorithm performed concurrently for the geometrical cells. The results are compliant with the ones for the verified predecessor model (**Figure 3.6**), which allows the safe verification of the current model.

## 4.3 Application Profiling & Code Optimization

### 4.3.1 A summary of the effects of the data preprocessing step, the speedup, the memory footprint reduction and the overall improved potential for the code optimization step it introduces

The data preprocessing step addresses the simulation resolution and costs tradeoff problem according to the algorithm of **Figure 3.3**. The data preprocessing algorithm adjusts the size of the geometrical cell so that the size of the 3D matrix that represents the solid tumor in silico is reduced while the corresponding simulation error that is introduced compared to the simulation execution for the original input size is maintained within a minimal tolerable window.

More specifically, as the size of the geometrical cell increases, each voxel of the 3D matrix that represents the solid tumor in silico encompasses a greater physical area and, as a result, the size of the 3D matrix and the simulation input size are reduced. However, as the size of the geometrical cell and the area that is encompassed in each voxel of the 3D tumor are increased, the number of biological cells that reside in each geometrical cell and in each equivalence class inside each geometrical cell are also increased (see **Figure 2.9**), which causes the simulation resolution to decrease and introduces an error to the outcomes of the simulation compared to those of the simulation execution for the original resolution and size for the geometrical cell.

The data preprocessing step was performed for each dataset patient and the corresponding set of simulation input files (.mhd and .raw). The results, with respect to the chosen scale factor for the size of the geometrical cell (i.e. GC scale factor) that optimally addresses the simulation resolution and costs tradeoff problem for each dataset patient, are summarized in **Tablre 4.5**. The chosen GC scale factor for Patient_15 and Patient_16 is equal to 2, while the chosen GC scale factor for Patient_17 is equal to 3.

The simulation input size reduction, caused by the data preprocessing induced increase of the geometrical cell size, results in the performance enhancement of the simulation execution, with respect both to the simulation execution time and to the memory footprint of the simulation execution.

The simulation execution time, analyzed in the times occupied by each scan of the oncosimulator algorithm (see **Figure 2.7**), is presented in the bar plots of **Figure 4.13** for each dataset patient and for GC scale factor values that correspond to the original input size (equal to 1) and to the optimal resolution adaptation of the data preprocessing step (equal to 2 for Patient_15 and Patient_16 and equal to 3 for Patient_17).

Reviewing **Figure 4.13**, the speedup of the simualtion execution for the input with increased geometrical cell size compared to the simualtion execution for the original input is evident, both in the total simulation execution time and in each individual simulation scans. The fourth, fifth and sixth scan that compose the simulation in the tissue level of biocomplexity take up most of the simulation execution time before the data preprocessing step. However, they are more favored by the input size reduction that the data preprocessing step introduces compared to the rest simulation scans, which causes a more fair workload balance in the simulation execution with the preprocessed inputs. This renders the data preprocessing step a crucial preparatory step before the code optimization with improved resources is performed, since a fair workload reduces the speedup limitations enforced by Amdahl's law (see **Equation 3.7**).

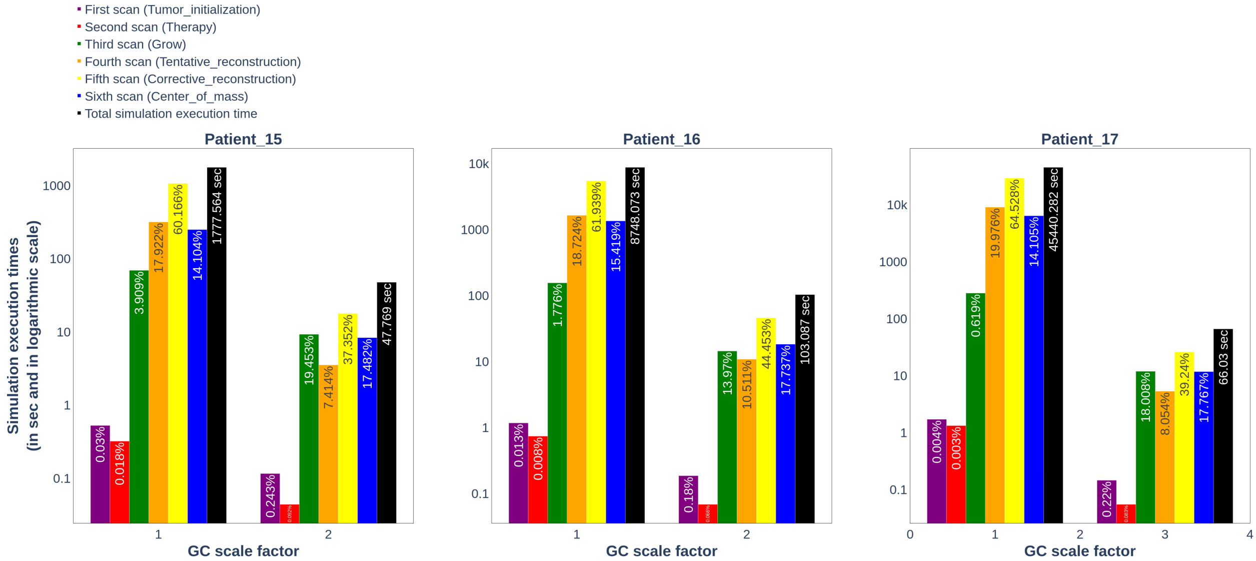


**Figure 4.13:** The bar plots that visualize the simulation execution times, both overall and analyzed in the individual simulation scans, before and after the data preprocessing step is applied for each dataset patient. The GC scale factor that is equal to 1 corresponds to the original input before the data preprocessing step is applied. The GC scale factor after the data preprocessing step is applied is equal to 2 for Patient_15 and Patient_16 and equal to 3 for Patient_17. As the size of the geometrical cell increases, the size of the 3D matrix that represents the solid tumor in silico, i.e. the simulation input size, decreases. As a result, the simulation execution time is also decreased. The corresponding speedup of the simulation execution is present in all of the individual simulation scans, but it is more noticeable in the simulation on the tissue level of biocomplexity (fourth, fifth and sixth scan), which renders the simulation workload balance more fair and improves the code optimization induced speedup prospects, which are limited according to Amdahl's law.

The data preprocessing induced speedup, both overall and for each individual simualtion scan, is displayed in detail for each dataset patient in the bar plot of **Figure 4.14**. The speedup is greater for the fourth, fifth and sixth scans, which compose the simulation on the tissue level of the biocomplexity. This is attributed to the fact that these scans simulate the biomechanical processes that adjust the shape and size of the tumor by performing many passes of the 3D matrix that represents the tumor in silico. As a result, they encompass more data dependencies and are greatly affected by the simulation input size reduction that the data preprocessing step introduces. More specifically, the greatest speedup is reported for the fourth scan and the fifth scan. The value of the speedup for the sixth scan is approximately half the value of the speedup for the fourth scan. The values of the speedup for the second scan and the third scan are approximately the same and noticeably lower than the values of the speedup for the simulation on the tissue level of biocomplexity (roughly by a factor of $10^{(s-1)}$ where $s$ is the GC scale factor). The value of the speedup for the first scan is the lowest, approximately half the value of the speedup for the third scan. The overall speedup is greater for Patient_17 that has a GC scale factor equal to 3. The overall speedups for Patient_15 and Patient_16, that have a GC scale factor equal to 2, are approximating, with the speedup for Patient_16 that has the greater original input size being higher.

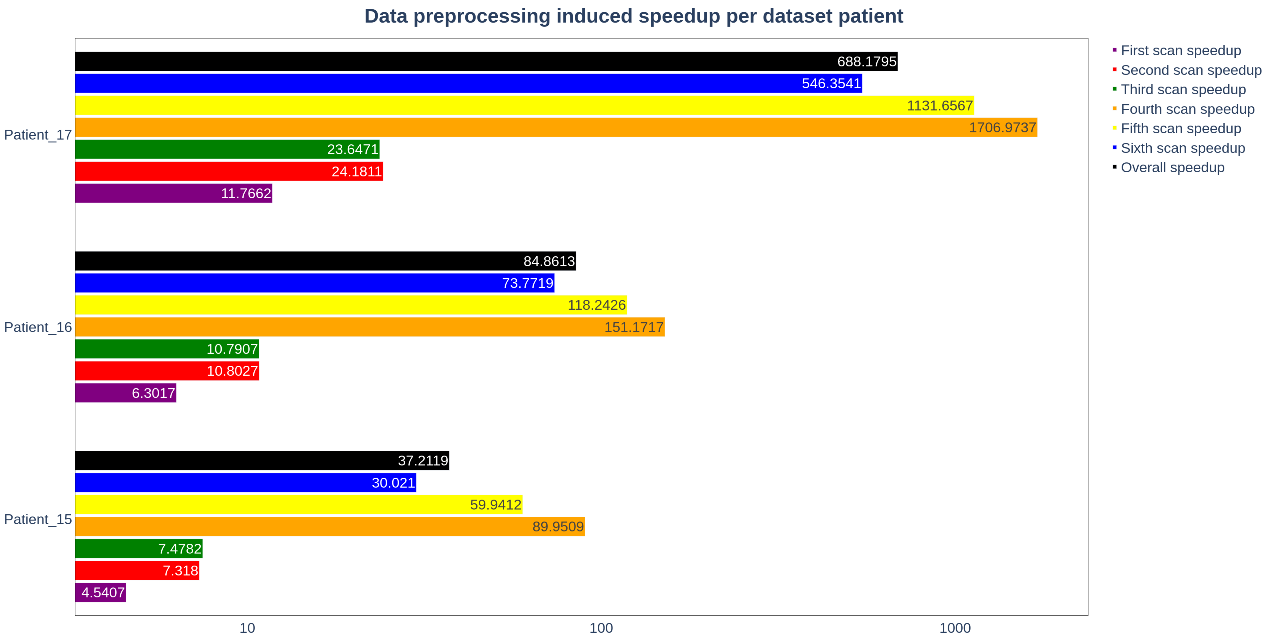


**Figure 4.14:** The bar plots that visualize the speedup for the simulation execution when data preprocessing is performed compared to the simulation execution when no data preprocessing is performed, both overall and for each individual simulation scan, for each dataset patient. The overall speedup is greater for Patient_17 that has a GC scale factor of 3. The overall speedups for Patient_15 and Patient_16, that have a GC scale factor of 2, are approximating, with the speedup for Patient_16 that has the greater original input size being higher. The simulation on the tissue level of biocomplexity performs multiple passes of the 3D matrix that represent the tumor in silico and, as a result, is the most favored by the data preprocessing induced simulation input size reduction. That fact is mirrored in the speedup values for the corresponding scans (fourth, fifth and sixth), which are noticeably greater compared to the speedup values for the rest of the scans.

The memory footprint of the simulation execution is presented in the bar plots of **Figure 4.13** for each dataset patient and for GC size scale factor values that correspond to the original input size (equal to 1) and to the optimal resolution adaptation of the data preprocessing step (equal to 2 for Patient_15 and Patient_16 and equal to 3 for Patient_17).

Reviewing **Figure 4.13**, the memory footprint reduction for the simulation execution for the input with increased geometrical cell size compared to the simualtion execution for the original input is evident. The memory footprint is determined by the simulation input size, which is reduced as the size of the geometrical cell is increased in the data preprocessing step. The memory footprint is recorded for the simulation execution, both before and after the data preprocessing is performed, using the valgrind massif heap profiler. The factor by which the memory footprint is reduced by the data preprocessing step is estimated as the division of the the size of the 3D matrix that represents the tumor in silico before the data preprocessing is performed (see **Table 4.1**) by the size of the 3D matrix that represents the tumor in silico after the data preprocessing is performed (see **Table 4.5**). This factor is equal to 4.5 for Patient_15 and Patient_16 and equal to 15.4 for Patient_17. Indeed, the greatest memory footprint reduction is recorded for Patient_17 that corresponds to a GC scale factor of 3, which is greater than the ones assigned for Patient_15 and Patient_16. Moreover, the expected reduction factors, as they are estimated according to the data of **Tables 4.1 and 4.5**, are approximately equal to the ones derived for the actual memory footprints that were recorded for each dataset patient using the massif tool.

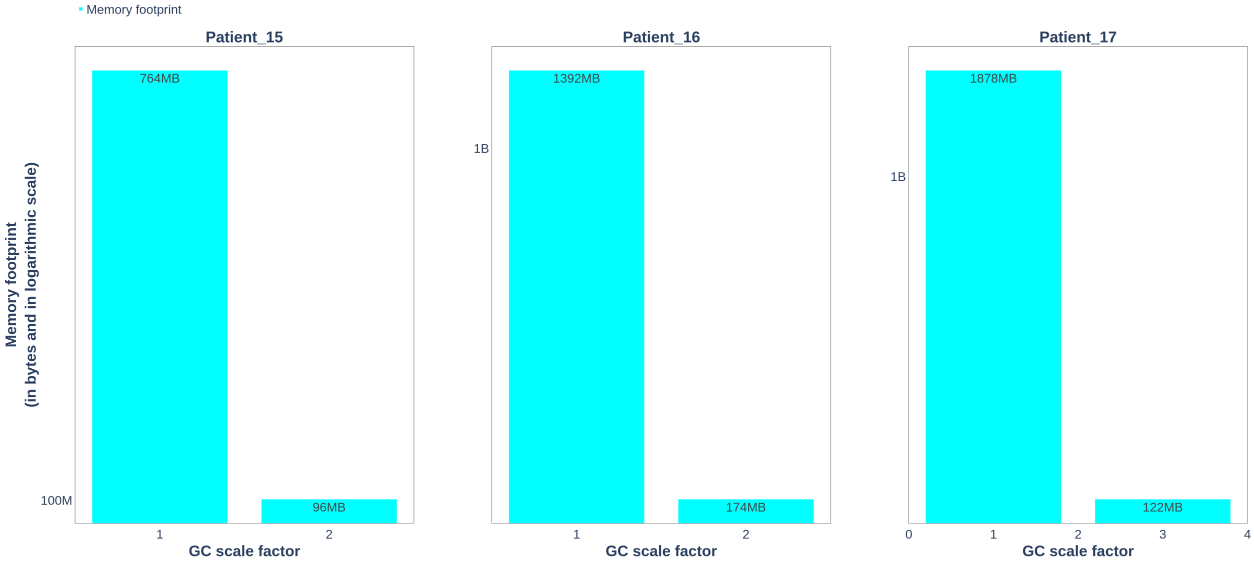


**Figure 4.15:** The bar plots that visualize the memory footprint of the simulation executions for each dataset patient, before and after the data preprocessing step is applied. The GC scale factor that is equal to 1 corresponds to the original input before the data preprocessing step is applied. The GC scale factor after the data preprocessing step is applied is equal to 2 for Patient_15 and Patient_16 and equal to 3 for Patient_17. As the size of the geometrical cell increases, the size of the 3D matrix that represents the solid tumor in silico, i.e. the simulation input size, decreases. As a result, the memory footprint of the simulation execution is also decreased. The memory footprint of the simulation executions is recorded using the valgrind massif heap profiler. The expected factor by which the memory footprint is reduced is estimated for each dataset patient as the division of the original input size by the input size after data preprocessing is applied. This factor is equal to 4.5 for Patient_15 and Patient_16 and equal to 15.4 for Patient_17. The massif recorded memory footprints are in compliance with these factors. The memory footprint of the simulation execution for Patient_17 is reduced more compared to the simulation executions for the rest of the dataset patients, since the GC scale factor has a greater value (equal to 3) for Patient_17.

The data preprocessing induced relative memory footprint reduction percentage is displayed in detail for each dataset patient in the bar plot of **Figure 4.16**. The memory footprint is reduced for each dataset patient by a percentage approximately equal to 90%. The relative memory footprint reduciton percentage is slightly higher (approximately 94%) for Patient_17, since the corresponding GC scale factor is equal to 3 and greater than the one for Patient_15 and Patient_16 (GC scale factor equal to 2). The relative memory footprint reduciton percentage is slightly lower (approximately 87%) for Patient_15 and Patient_16.

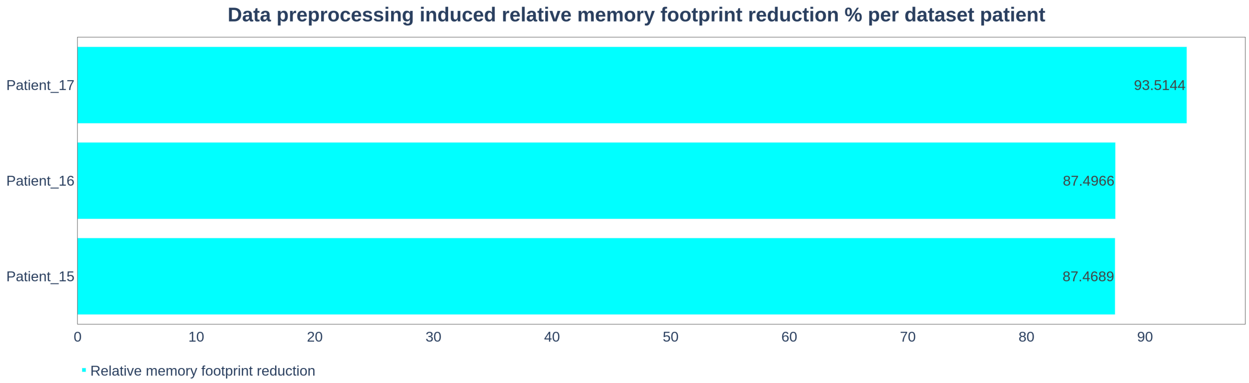


**Figure 4.16:** The bar plots that visualize the data preprocessing induced relative memory footprint reduction percentage for each dataset patient. The relative memory footprint reduction percentage is approximately equal to 87% for Patient_15 and Patient_16, that are characterized by a GC scale factor of 2. The relative memory footprint reduction percentage is slightly higher (approximately equal to 94%) for Patient_17, that is characterized by a greater GC scale factor value (equal to 3).

As mentioned previously, the data preprocessing step favors more the fourth, fifth and sixth scan that compose the simulation in the tissue level of biocomplexity, since it reduces the simulation input size and these scans require more passes of the 3D matrix that represents the tumor in silico. As a result, the execution time required by these scans is reduced more by the data preprocessing step, compared to the execution time reduction that refers to the rest of the scans (see **Figures 4.13 and 4.14**).

Consequently, the percentage of the execution time that the biomechanical simulation part of scans four, five and six represents is limited after the data preprocessing step is performed, rendering the workload balance more fair. More importantly, the speedup prospects of the code optimization that follows, which focuses on the concurrent execution of the third scan for the geometrical cells of the 3D tumor, are improved, since the maximum speedup is limited according to Amdahl's law by the part of the task that does not benefit from the improved resources (see **Equation 3.7**), here the fourth, fifth and sixth scans.

All in all, except for the speedup and the memory footprint reduction that the data preprocessing step immediately introduces, it also indirectly helps the speedup prospects that the code optimization that follows introduces by reducing the percentage of the workload that does not benefit from the improved resources, i.e. the multiple threads that execute computations concurrently. This indirect effetc of the data preprocessing step is presented in **Figure 4.17**, which displays the percentage of execution time that is attributed to each individual simulation scan for each dataset patient, before and after the data preprocessing step is applied. The part $p$ that benefits from the improved resources of the code optimization to follow corresponds to the third scan of the simulation and the maximum speedup $S$ allowed according to Amdahl's law is given by **Equation 3.7**. After the data preprocessing step is applied, $p$ is increased, which causes $S$ to be increased too.

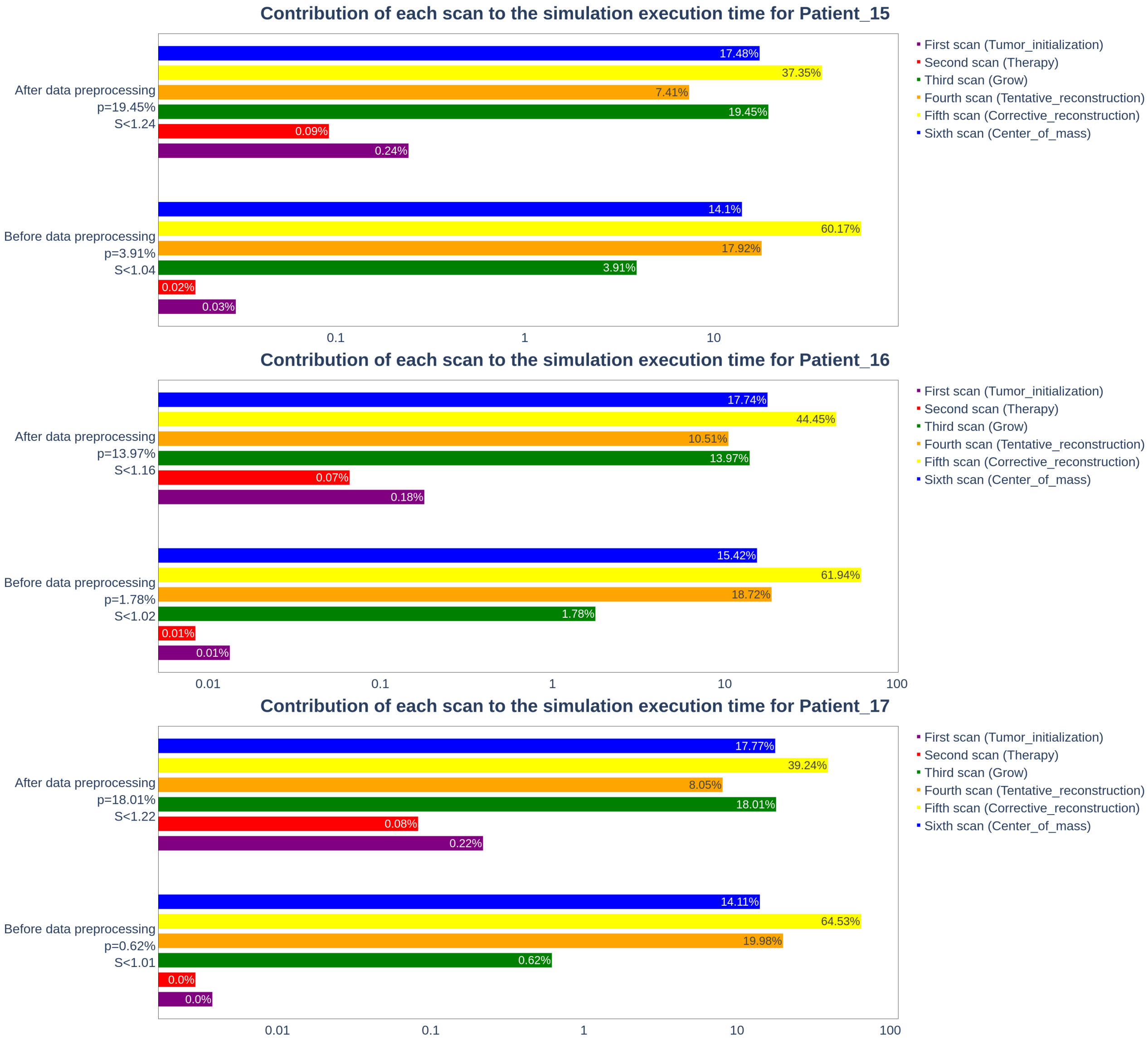


**Figure 4.17:** The percentages of execution time that are attributed to each individual simulation scan for each dataset patient, before and after the data preprocessing step is applied, which indicate the indirect effect of the data preprocessing step to the maximum speedup allowed according to Amdahl's law. After the data preprocessing step is applied, the part $p$ of the simulation that benefits from the improved resources introduced in the code optimization step that follows is increased, which causes the maximum speedup $S$, according to Amdahl's law and **Equation 3.7**, to be increased too.

### 4.3.2 The simulation execution callgraph and the costs per function, with respect to the memory accesses and to the CPU cycles count

The distribution of the program execution cost to the various parts of the source code is an important step before any code optimization attempt, with ultimate goal the execution acceleration, is made. This type of profiling for the program execution, with respect to the costs that each major part of the source code introduces and to the percentage of the overall execution cost it represents, provides crucial information concerning the structure of the execution and the source code parts that pose the greater performance inhibitors and that need to be addressed within an optimization plan context, so that the performance is enhanced, both for the respective source code part and consequently for the overall execution.

Here, this type of source code sections analytical cost profiling is performed for the executions of the Nephroblastoma Oncosimulator for each dataset patient, after the data preprocessing step is performed (i.e. with the inputs of **Tabel 4.5**), using the callgrind tool that is provided by the valgrind framework. The profiling results, which include the structure of the execution in the form of the hierarchy of source code function calls and the execution costs per function, are visualized in the callgraphs of **Figures 4.18, 4.19 and 4.20** for Patient_15, Patient_16 and Patient_17, respectively. These callgraphs were produced by the KCachegrind tool, which is used for the visualization of the ouputs of the callgrind profiler [80].

Each callgraph is a directed acyclic graph, with each node corresponding to a source code function and the directed edges corresponding to function calls. More specifically, the parent function calls the child function and the number of calls is recorded as the label of the corresponding edge. Moreover, each node has a label that expresses the percentage of the execution cost that corresponds to the specific source code function, with respect to the CPU cycles count. These cost percentages are inclusive, meaning that the recorded cost for each function includes the cost of the functions called by it. This is why the parent nodes/functions have a greater cost pecentage compared to their corresponding child nodes/functions.

The callgraphs of **Figures 4.18-4.20** have a similar structure, with only the functions that have a considerable contribution to the execution cost being included, in order for the graphs to be more sparse and readable. The `main` function, which corresponds to the total execution cost, calls the `_therapy_combi_drug_admin_points` function, which performs the simulation evolution with a loop of discrete time steps. Within each time step, the consecutive scans of the 3D matrix that represents the tumor in silico are performed with consecutive calls of the corresponding functions, `advance_cycle`, `tentative_reconstruction`, `corrective_reconstruction` and `center_of_mass`, within three nested loops.

The `_therapy_combi_drug_admin_points` function is the wrapper function that coordinates the calls to the major simulation components for the evolution of the simulation. As such, it encompasses most of the execution cost, approximately 99.9% of the total cost.

The `advance_cycle` function performs the second and third scan for the current time step and geometrical cell of the 3D tumor matrix. It is called by the `_therapy_combi_drug_admin_points` function for each simulation time step and for each geometrical cell of the 3D tumor matrix. It is responsible for approximately $12-18$ % of the total simulation execution cost, which is in compliance with the percentages of **Figure 4.13** (slightly underestimated).

The `tentative_reconstruction` function performs the fourth scan for the current time step and geometrical cell of the 3D tumor matrix. It is called by the `_therapy_combi_drug_admin_points` function for each simulation time step and for each geometrical cell of the 3D tumor matrix that corresponds to tumor tissue and has an excess of biological cells. It is responsible for approximately $10 - 13$ % of the total simulation execution cost, which is in compliance with the percentages of **Figure 4.13** (slightly overestimated).

The `tentative_reconstruction` function calls the `sumd` and `find_appropriate_neighbour` functions, in order to count the excess of biological cells and find a neighboring geometrical cell that can support an income of biological cells. The `createPos` and `move_to_neighb` functions are called so that the part of the excess of biological cells that can be supported by the chosen neighbor is unloaded. The process is repeated until the excess of biological cells is unloaded to the neighboring geometrical cells or until no more income of biological cells can be supported by the neighboring geometrical cells. If the excess remains and no neighboring geometrical cell that can support an income of biological cells exists, a new geometrical cell is created and incorporated in the 3D tumor matrix using the `push_expand` function.

The `corrective_reconstruction` function performs the fifth scan for the current time step and geometrical cell of the 3D tumor matrix. It is called by the `_therapy_combi_drug_admin_points` function for each simulation time step and for each geometrical cell of the 3D tumor matrix that corresponds to tumor tissue and has a lack of biological cells. It is responsible for approximately $46 - 53$ % of the total simulation execution cost, which is in compliance with the percentages of **Figure 4.13** (slightly overestimated).

The `corrective_reconstruction` function calls the `total_population` and `find_appropriate_neighbour` functions, in order to count the total number of biological cells and find a neighboring geometrical cell that can support an income of biological cells. The `createPos` and `move_to_neighb` functions are called so that the part of the biological cells that can be supported by the chosen neighbor is unloaded. The process is repeated until all the biological cells are unloaded and the current geometrical cell is deleted from the 3D tumor matrix using the `push_shrink` function.

The `center_of_mass` function performs the sixth scan for the current time step and geometrical cell of the 3D tumor matrix. It is called by the `_therapy_combi_drug_admin_points` function for each simulation time step and for each geometrical cell of the 3D tumor matrix that corresponds to tumor tissue. It is responsible for approximately $13 - 14$ % of the total simulation execution cost, which is in compliance with the percentages of **Figure 4.13** (slightly underestimated).

The `center_of_mass` function ensures that th area of the 3D tumor matrix that corresponds to tumor tissue is maintained compact after the changes introduced by the `tentative_reconstruction` and `corrective_reconstruction` functions. More specifically, if a lone tumor tissue geometrical cell exists, it is shifted toward the center of mass of the 3D tumor matrix. One the other hand, if a hole in the tumor tissue of the 3D matrix exists, then it is deleted from the tumor by calling the `push_shrink` function for it (see callgraphs of **Figures 4.19 and 4.20**).

Finally, the `find_appropriate_neighbour` function, which is called both by the `tentative_reconstruction` function and by the `corrective_reconstruction` function, calls the `total_population` function, in order to count the number of biological cells in each neighbor geometrical cell for the current geometrical cell.

| Source code function | Scan |
|---|---|
| therapy_combi_drug_admin_points | $N/A$ |
| advance_cycle | 3 |
| tentative_reconstruction | 4 |
| corrective_reconstruction | 5 |
| center_of_mass | 6 |
| sumd | 4 |
| find_appropriate_neighbour | 4, 5 |
| createPos | 4, 5 |
| move_to_neighb | 4, 5 |
| push_expand | 4 |
| push_shrink | 5, 6 |
| total_population | 4, 5, 6 |

**Table 4.6:** The primary Nephroblastoma Oncosimulator source code functions and the corresponding simulation scans that they are referenced in.

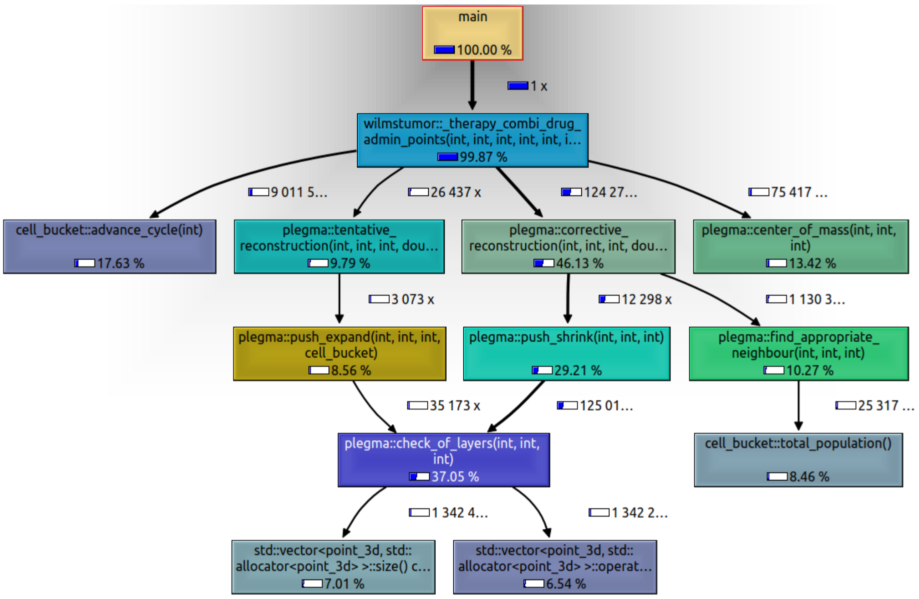

main
100.00 %
1 x
wilmstumor::_therapy_combi_drug_admin_points(int, int, int, int, int, i…
99.87 %
9 011 5…
26 437 x
124 27…
75 417 …
cell_bucket::advance_cycle(int)
17.63 %
plegma::tentative_reconstruction(int, int, int, dou…
9.79 %
plegma::corrective_reconstruction(int, int, int, dou…
46.13 %
plegma::center_of_mass(int, int, int)
13.42 %
3 073 x
12 298 x
1 130 3…
plegma::push_expand(int, int, int, cell_bucket)
8.56 %
plegma::push_shrink(int, int, int)
29.21 %
plegma::find_appropriate_neighbour(int, int, int)
10.27 %
35 173 x
125 01…
25 317 …
plegma::check_of_layers(int, int, int)
37.05 %
cell_bucket::total_population()
8.46 %
1 342 4…
1 342 2…
std::vector<point_3d, std::allocator<point_3d> >::size() c…
7.01 %
std::vector<point_3d, std::allocator<point_3d> >::operat…
6.54 %

**Figure 4.18:** The callgraph derived for the execution of the Nephroblastoma Oncosimulator for the data preprocessed input for Patient_15, as it was profiled with the callgrind tool and visualized with the KCachegrind tool. Each node corresponds to a source code function, while the directed edges correspond to function calls from the parent function to the child function. The edge labels indicate the number of calls, while the node labels indicate the constribution of the corresponding function to the overall execution cost. These costs are inclusive, meaning that the parent nodes encompass the cost of their corresponding child nodes.

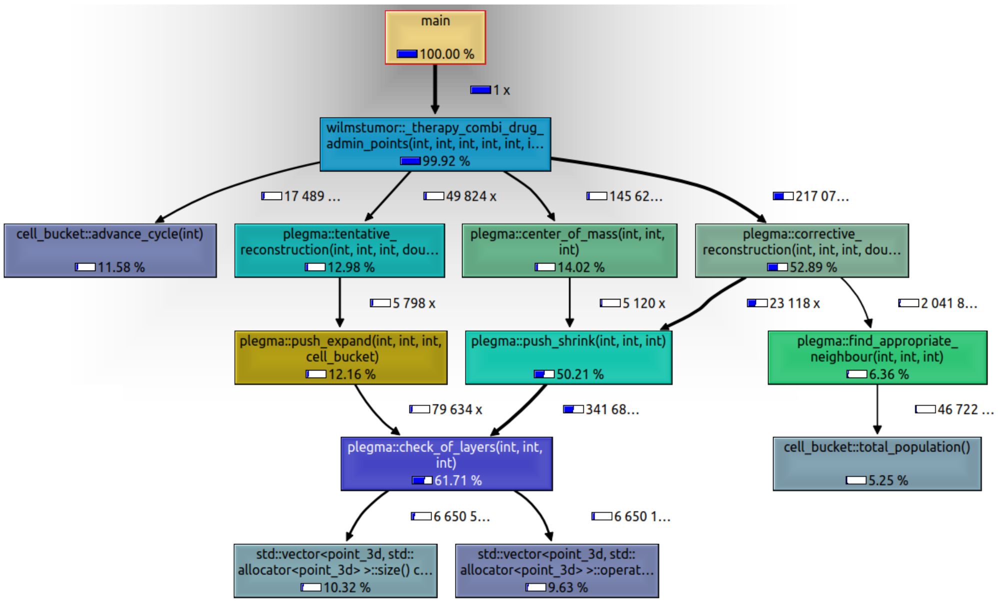

main
100.00 %
1 x
wilmstumor::_therapy_combi_drug_admin_points(int, int, int, int, i...
99.92 %
17 489 ...
49 824 x
145 62...
217 07...
cell_bucket::advance_cycle(int)
11.58 %
plegma::tentative_reconstruction(int, int, int, dou...
12.98 %
plegma::center_of_mass(int, int, int)
14.02 %
plegma::corrective_reconstruction(int, int, int, dou...
52.89 %
5 798 x
5 120 x
23 118 x
2 041 8...
plegma::push_expand(int, int, int, cell_bucket)
12.16 %
plegma::push_shrink(int, int, int)
50.21 %
plegma::find_appropriate_neighbour(int, int, int)
6.36 %
79 634 x
341 68...
46 722 ...
plegma::check_of_layers(int, int, int)
61.71 %
cell_bucket::total_population()
5.25 %
6 650 5...
6 650 1...
std::vector<point_3d, std::allocator<point_3d> >::size() c...
10.32 %
std::vector<point_3d, std::allocator<point_3d> >::operat...
9.63 %

**Figure 4.19:** The callgraph derived for the execution of the Nephroblastoma Oncosimulator for the data preprocessed input for Patient_16, as it was profiled with the callgrind tool and visualized with the KCachegrind tool. Each node corresponds to a source code function, while the directed edges correspond to function calls from the parent function to the child function. The edge labels indicate the number of calls, while the node labels indicate the constribution of the corresponding function to the overall execution cost. These costs are inclusive, meaning that the parent nodes encompass the cost of their corresponding child nodes.

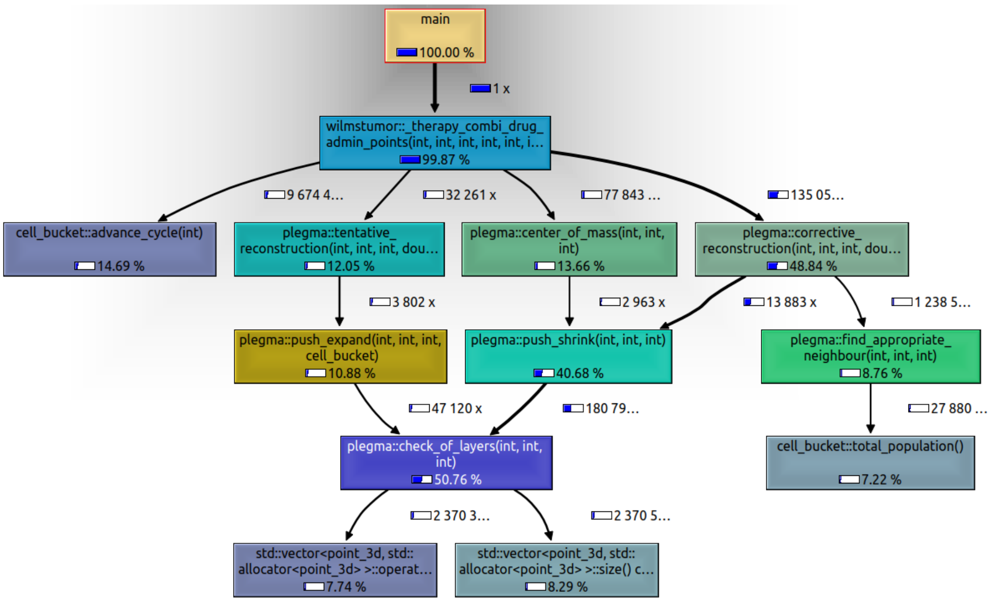

main
100.00 %
1 x
wilmstumor::_therapy_combi_drug_admin_points(int, int, int, int, i...
99.87 %
9 674 4...
32 261 x
77 843 ...
135 05...
cell_bucket::advance_cycle(int)
14.69 %
plegma::tentative_reconstruction(int, int, int, dou...
12.05 %
plegma::center_of_mass(int, int, int)
13.66 %
plegma::corrective_reconstruction(int, int, int, dou...
48.84 %
3 802 x
2 963 x
13 883 x
1 238 5...
plegma::push_expand(int, int, int, cell_bucket)
10.88 %
plegma::push_shrink(int, int, int)
40.68 %
plegma::find_appropriate_neighbour(int, int, int)
8.76 %
47 120 x
180 79...
27 880 ...
plegma::check_of_layers(int, int, int)
50.76 %
cell_bucket::total_population()
7.22 %
2 370 3...
2 370 5...
std::vector<point_3d, std::allocator<point_3d> >::operat...
7.74 %
std::vector<point_3d, std::allocator<point_3d> >::size() c...
8.29 %

**Figure 4.20:** The callgraph derived for the execution of the Nephroblastoma Oncosimulator for the data preprocessed input for Patient_17, as it was profiled with the callgrind tool and visualized with the KCachegrind tool. Each node corresponds to a source code function, while the directed edges correspond to function calls from the parent function to the child function. The edge labels indicate the number of calls, while the node labels indicate the constribution of the corresponding function to the overall execution cost. These costs are inclusive, meaning that the parent nodes encompass the cost of their corresponding child nodes.

The callgrind tool was executed with the Nephroblastoma Oncosimulator for each dataset patient with the `--caches-sim` option activated. This means that a two level cache hierarchy was simulated during the profiling and the memory accesses per function, to the first and last level of the cache hierarchy, as well as to the main memory, were calculated. That way, the distribution of the memory access induced costs to the source code functions is calculated. The percentage of hits and misses to the cache hierarchy per source code function, as well as the CPU cycles count percentage of the overall cycles per source code function, are visualized in **Figures 4.21-4.23** for the three dataset patients.

The $Ir$, $Dr$ and $Dw$ percentages refer to the hits to the first level cache, with respect to instruction reads, data reads and data writes, respectively. The $I1mr$, $D1mr$ and $D1mw$ percentages refer to the misses to the first level cache, with respect to instruction reads, data reads and data writes, respectively. The $ILmr$, $DLmr$ and $DLmw$ percentages refer to the misses to the last level cache, which correspond to accesses to the main memory, with respect to instruction reads, data reads and data writes, respectively. Each instruction read hit corresponds to one CPU cycle, for the execution of the instruction. Each access to the last level cache corresponds to 10 CPU cycles, while each access to the main memory corresponds to 100 CPU cycles. That way, the CPU cycles count is estimated according to **Equation 3.4**.

The memory accesses and CPU cycles count percentages per function that are displayed in **Figures 4.21-4.23** for the three dataset patients are not inclusive, meaning that, unlike the costs of the functions of nodes of the callgraphs of **Figures 4.18-4.18**, they fully refer to the corresponding function and the calls to other functions are not taken into consideration. That way, the exact source code lines that contribute to the CPU cycle count and memory accesses costs become clear.

The `advance_cycle` function contributes to the total number of CPU cycles by a percentage of approximately $10 - 15$ %, which is in compliance with the third scan induced costs in the callgraphs of **Figures 4.18-4.18**.

The contribution of the fourth and fifth scans to the CPU cycles count, via the `tentative_reconstruction` and the `corrective_reconstruction` function calls, is distributed to the functions that are called within these scans (see **Table 4.6**), `find_appropriate_neighbour`, `sumd`, `move_to_neighb`, `createPos` and especially to `total_population`. The `corrective_reconstruction` function is called many times, since the simulation is performed for therapy application that shrinks the tumor, which results in a noticeable CPU cycles count direct cost for this function, while the `tentative_reconstruction` function is not called many times and as a result the corresponding CPU cycles count direct cost for this function is minimal.

The `center_of_mass` function that is called in the sixth scan has a noticeable direct contribution to the total CPU cycles count. It also calls the `total_population` function, which explains the great constribution that this function has to the total number of CPU cycles, since it is constantly present throughout the simulation on the tissue level of biocomplexity (fourth, fifth and sixth scans).

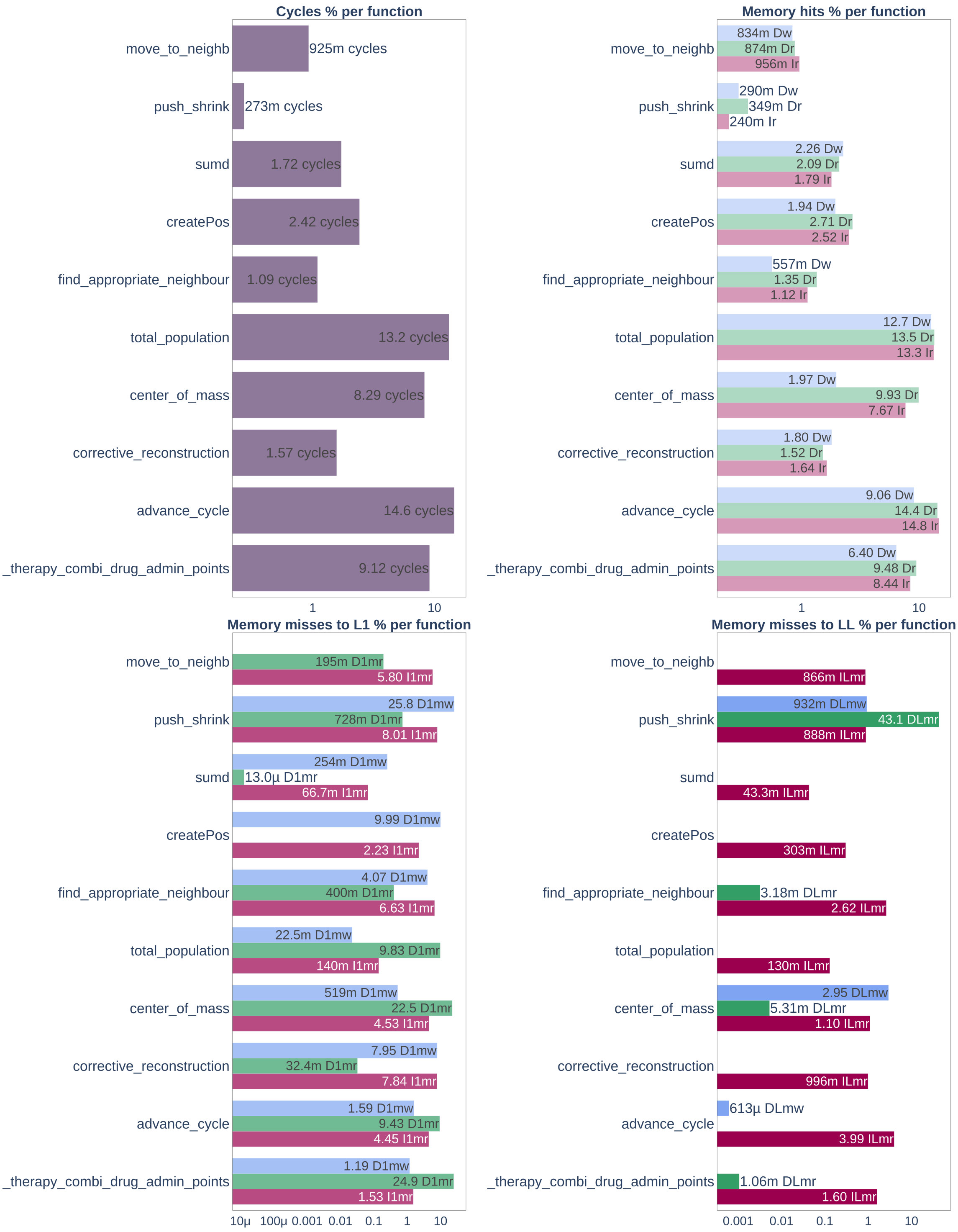


**Figure 4.21:** The CPU cycles count and memory percentages per source code function for the execution of the Nephroblastoma Oncosimulator with the data preprocessed inputs for Patient_15, as they were calculated using the callgrind profiler with the cache hierarchy simulation activated. The `advance_cycle` function (third scan) contributes to the total number of CPU cycles by a percentage of 14.6%. The `total_population` function, which is called by the `tentative_reconstruction` (fourth scan), `corrective_reconstruction` (fifth scan) and `center_of_mass` (sixth scan) has the second biggest cost, with a contribution of 13.2% to the CPU cycles count. The `_therapy_combi_drug_admin_points` and the `center_of_mass` functions follow with a contribution percentage of 9.12% and 8.29%, respectively. The rest of the functions do not contribute greatly to the execution cost.

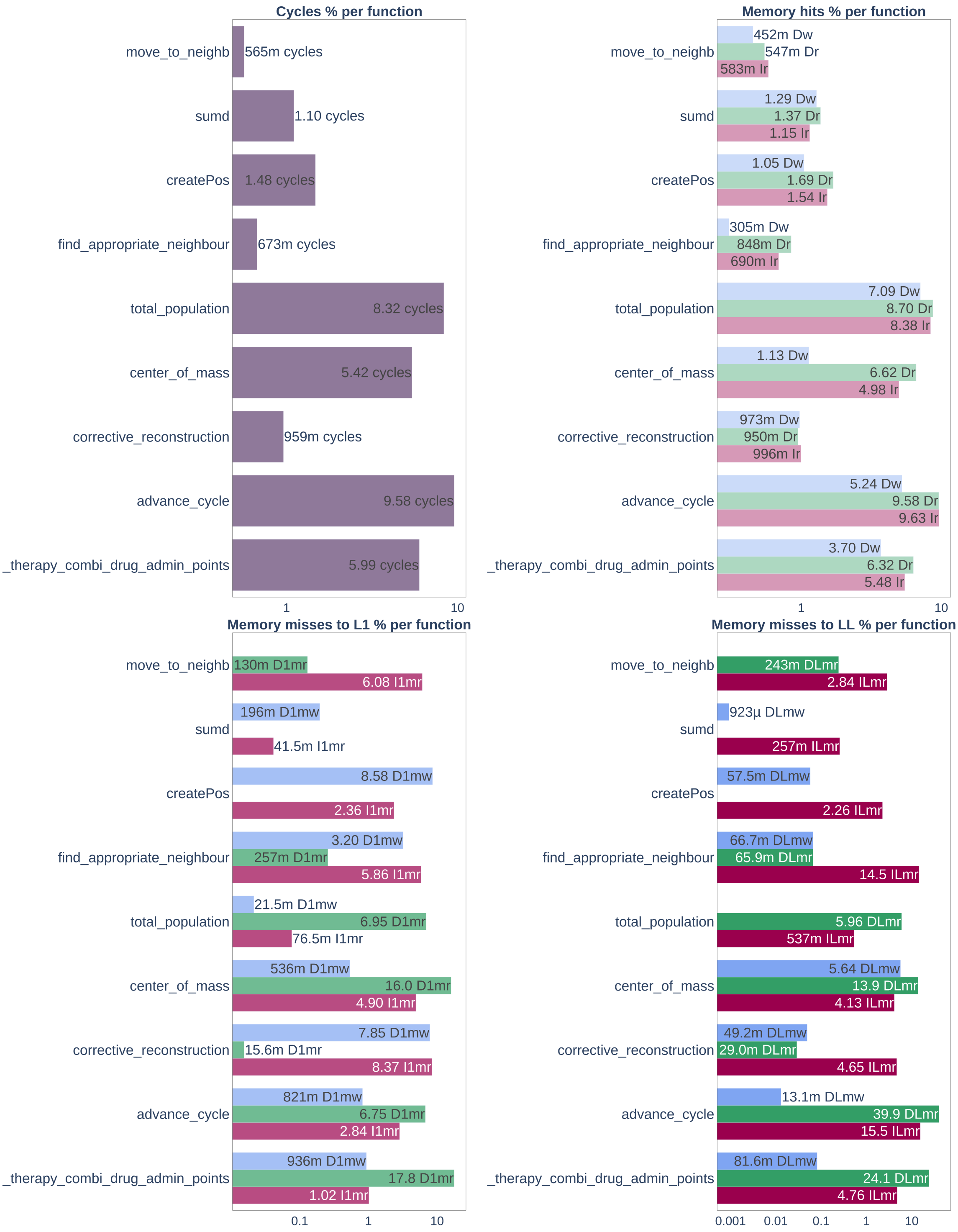


**Figure 4.22:** The CPU cycles count and memory percentages per source code function for the execution of the Nephroblastoma Oncosimulator with the data preprocessed inputs for Patient_16, as they were calculated using the callgrind profiler with the cache hierarchy simulation activated. The `advance_cycle` function (third scan) contributes to the total number of CPU cycles by a percentage of 9.58%. The `total_population` function, which is called by the `tentative_reconstruction` (fourth scan), `corrective_reconstruction` (fifth scan) and `center_of_mass` (sixth scan) has the second biggest cost, with a contribution of 8.32% to the CPU cycles count. The `_therapy_combi_drug_admin_points` and the `center_of_mass` functions follow with a contribution percentage of 5.99% and 5.42%, respectively. The rest of the functions do not contribute greatly to the execution cost.

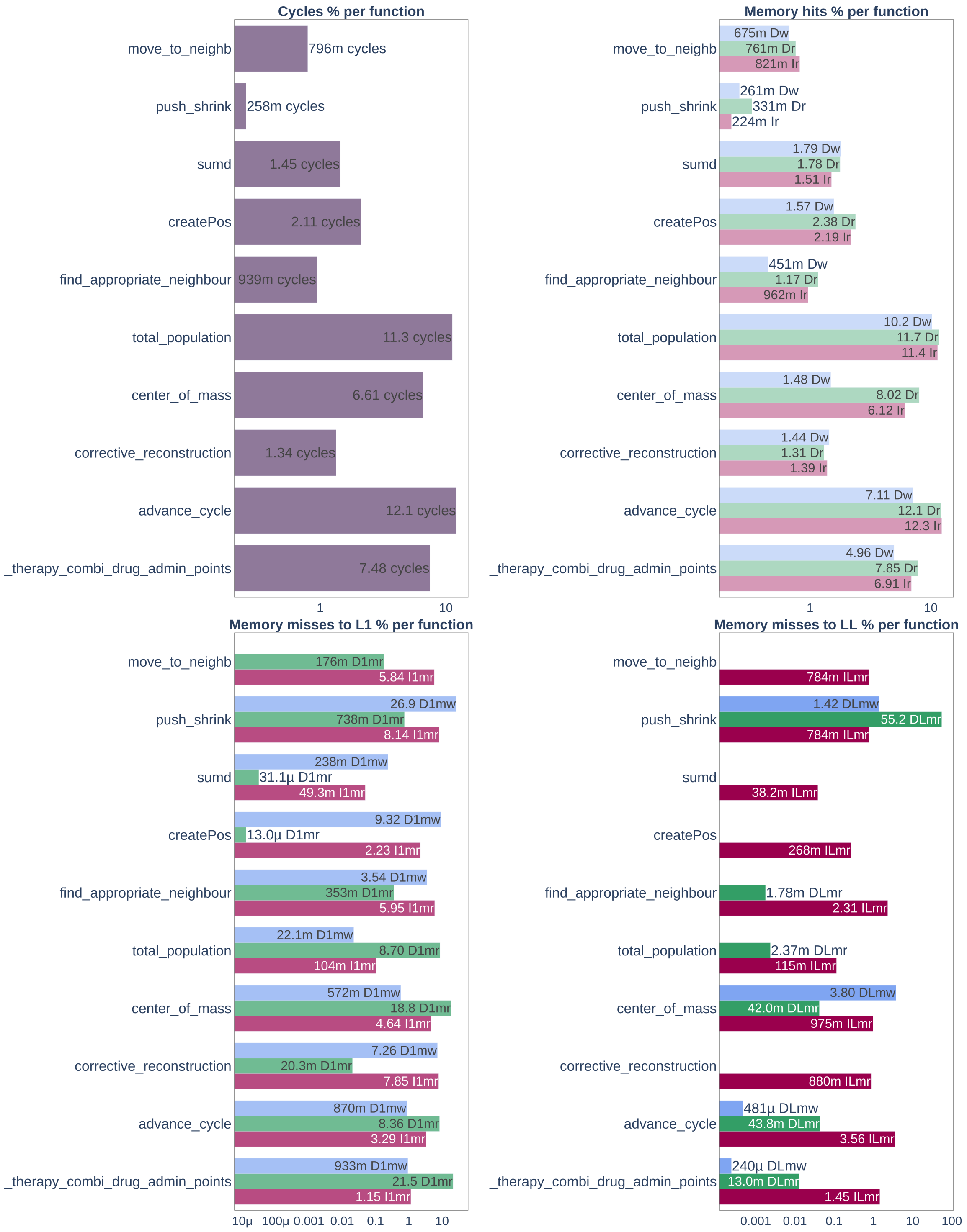


**Figure 4.23:** The CPU cycles count and memory percentages per source code function for the execution of the Nephroblastoma Oncosimulator with the data preprocessed inputs for Patient_17, as they were calculated using the callgrind profiler with the cache hierarchy simulation activated. The `advance_cycle` function (third scan) contributes to the total number of CPU cycles by a percentage of 12.1%. The `total_population` function, which is called by the `tentative_reconstruction` (fourth scan), `corrective_reconstruction` (fifth scan) and `center_of_mass` (sixth scan) has the second biggest cost, with a contribution of 11.3% to the CPU cycles count. The `_therapy_combi_drug_admin_points` and the `center_of_mass` functions follow with a contribution percentage of 7.48% and 6.61%, respectively. The rest of the functions do not contribute greatly to the execution cost.

## 4.4 Clinical Adaptation

### 4.4.1 A summary of the clinical adaptation executions

A series of executions of the clinical adaptation algorithm of **Figure 3.8** was performed, for each dataset patient and with varying values for the $N$ parameter, i.e. the parameter that defines the number of virtual patients (VPs) to be executed in parallel. The patient input files that were used are the ones that correspond to the optimal simulation resolution adaptation, as they were estimated in the data preprocessing step (see **Table 4.5**).

The parameter that was adapted for each VP, according to the serial exploration algorithm of **Figure 3.10**, was the total $CKR$ parameter, i.e. the parameter that expresses the total cell kill ratio of the administered therapy. The total $CKR$ parameter is analyzed into two distinct $CKR$ parameters, the cell kill ratio for the Actinomycin drug ($CKR\_ACT$) and the cell kill ratio for the Vincristine drug ($CKR\_VCR$), according to the following equations.

$$
\begin{aligned}
CKR\_ACT &= \frac{3}{5} \cdot CKR \\
CKR\_VCR &= \frac{2}{5} \cdot CKR
\end{aligned}
$$

The rest of the Nephroblastoma Oncosimulator input parameters were assigned values from the uniform distribution and within a window that allowed a deviation of up to 50% from their corresponding reference value.

The tolerable error window for the serial exploration of the total $CKR$ parameter value for each VP was set to 5% and was considered with respect to the tumor volume reduction, meaning that, for each VP, the value for the total $CKR$ parameter was adjusted so that the simulated tumor volume reduction deviates from the actual physical tumor volume reduction by a percentage of up to 5%.

The results that are presented here correspond to two distinct executions of the clinical adaptation algorithm of **Figure 3.8** for each dataset patient, one for 20 virtual patients ($N = 20$) and one for 200 virtual patients ($N = 200$). The attributes of the six clinical adaptation algorithm executions, two for each of the three dataset patients, are summarized in **Table 4.7**.

The first three rows of **Table 4.7** refer to the input files of the simulations that are executed for the clinical adaptation realization for each corresponding dataset patient. These input files are defined unambiguously by the dataset patient to be simulated, accompanied by a scale factor and padding factor set that determines the simulation resolution as described in the data preprocessing step. Here, the scale factor and padding factor set that is utilized for each dataset patient is the one that leads to the optimal resolution and error tradeoff for the simulation. Subsequently, the dataset patient that is simulated, characterized by the set of the scale factor and the padding factor, defines the execution time required by a single simulation (see the results of the data preprocessing step in **Table 4.5**).

The fourth row of **Table 4.7** defines the number of virtual patients that were executed for the realization of the corresponding clinical adaptation, which here is either 20 or 200. The definition of the clinical adaptation algorithm execution is completed by the definition of the distribution of the input parameters (the uniform distribution with a deviation window of 50% here), the parameter to be explored serially for each virtual patient (the total $CKR$ parameter here) and the criterion by which the adaptation is evaluated (the tumor volume reduction here). The tolerable adaptation error is documented in the fifth row of **Table 4.7** and here it is equal to 5% for all the clinical adaptation executions.

The first six rows of **Table 4.7** refer to the definition of the algorithm execution, while the remaining four rows summarize the characteristics of the outcome of the execution.

The eighth row of **Table 4.7** records the total number of executed simulations for each clinical adaptation execution, i.e. the sum of the number of serial simulation executions per virtual patient within the context of the clinical adaptation execution.

The seventh row of **Table 4.7** documents the total execution time for each clinical adaptation execution, which depends on the maximum time between the virtual patient executions, assuming that all the virtual patients are executed in parallel. More specifically, since the virtual patients are being executed in parallel, the total clinical adaptation execution time is approximately equal to the maximum number of serial simulation executions between the virtual patients, mutliplied by the execution time of a single simulation.

The number of serial simulation executions, i.e. the number of values that are explored for the adaptation parameter (i.e. the total $CKR$ parameter) before its value is adjusted so that the adaptation criterion is satisfied, differs between the virtual patients, depending on the values for the rest of the simulation input parameters. However, when the input parameters get their values from the uniform distribution, there is no reason to assume big deviation of this number between the distinct virtual patients.

Assuming that all the virtual patients are executed in parallel, the maximum number of serial simulation executions between the virtual patients of a clinical adaptation execution can be roughly estimated as the integer part of the division of the total clinical adaptation execution time by the time required for a single simulation execution. This estimate is diplayed in the ninth row of **Table 4.7**. When the computational cores are limited and as the number of the virtual patients $N$ increases, not all virtual patients are executed in parallel and the same metric provides an overestimated value for the maximum number of serial simulation executions between the virtual patients.

Finally, the average number of serial simulation executions between the virtual patients of the clinical adaptation execution is estimated as the integer part of the division of the total number of executed simulations by the number of virtual patients $N$. This estimate is documented in the tenth row of **Table 4.7**. It can be safely assumed that the displayed values are fairly accurate, since the simulation input parameters are assigned values from the uniform distribution, meaning that there are not many outliers in the set of numbers of serial simulation executions per virtual patient.

Reviewing the summary of the series of clinical adaptation executions that were performed for each dataset patient and with a varying value for the number of virtual patients $N$, as it is displayed in **Table 4.7**, the following key observations are made:

- For the executions with $N = 200$, the estimated value for the maximum number of serial executions is way higher than the estimated value for the average number of serial executions, indicating that the maximum value is overestimated and that the computational cores are not enough for the parallel execution of all the virtual patients.
- As the number of virtual patients $N$ increases, so does the estimated value for the average number of serial executions. This is indicative of the fact that clinical adaptation executions with more virtual patients are more probable to contain serial simulation executions that require an exploration of more values for the adaptation parameter until the adaptation criterion is met. This can be attributed to the fact that more virtual patient executions correspond to a potentially wider range of values for the joint distribution of the simulation input parameters.
- As the number of virtual patients $N$ increases, so does the total number of executed simulations, which is a characteristic that is inherent to the clinical adaptation algorithm, but is also attributed to the corresponding increase in the estimated value for the average number of serial executions, which is indicative of the number of values explored for the adaptation parameter.
- As the number of virtual patients $N$ increases, so does the total clinical adaptation execution time. That is mainly because the number of computational cores is not adequate for the parallel execution of all the virtual patients, but can also be attributed to the corresponding increase in the estimated value for the average number of serial executions.

| **Patient** | 15 | 15 | 16 | 16 | 17 | 17 |
|---|---|---|---|---|---|---|
| **scale** | 2 | 2 | 2 | 2 | 3 | 3 |
| **padding** | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| **N** | 20 | 200 | 20 | 200 | 20 | 200 |
| **Tolerable error** | 5% | 5% | 5% | 5% | 5% | 5% |
| **Single simulation execution time (*sec*)** | 47.77 | 47.77 | 103.09 | 103.09 | 66.03 | 66.03 |
| **Clinical adaptation execution time (*sec*)** | 206.97 | 3972.89 | 1185.96 | 6028.76 | 280.9 | 4719.75 |
| **Total number of executed simulations** | 50 | 1115 | 72 | 822 | 51 | 1193 |
| **Maximum number of simulations per VP (rough estimate)** | 4 | 83 | 11 | 58 | 4 | 71 |
| **Average number of simulations per VP (rough estimate)** | 2 | 5 | 3 | 4 | 2 | 5 |

**Table 4.7:** The summary of the clinical adaptation executions. The executions were performed for each dataset patient with a varying value for the number of virtual patients $N$ ($N = 20$ and $N = 200$). The simulation input files that were utilized are the ones that ensure the optimal resolution error tradeoff, unambiguously defined by the set of scale and padding factors (see the data preprocessing step). The adaptation parameter is the total $CKR$, while the rest of the simulation input parameters get values from the uniform distribution with a deviation window of 50%. The tolerable error is 5% and is computed with respect to the tumor volume reduction. The maximum number of serially executed simulations is estimated as the integer part of the division of the clinical adaptation execution time by the execution time of a single simulation (overestimated when not all virtual patients are executed in parallel). The average number of serially executed simulations is estimated as the integer part of the division of the total executed simulations by $N$. As $N$ increases so does the number of executed simulations, both total and serial within the context of each virtual patient, as well as the overall clinical adaptation execution time.

### 4.4.2 The distribution of the simulation input parameters for each clinical adaptation execution

The clinical adaptation algorithm of **Figure 3.8** is based on the concept of creating a series of virtual patients for each individual actual patient. Each virtual patient corresponds to a distinct representation of the actual physical patient in silico, which is unique with respect to the joint distribution of the values for the simulation input parameters. Here each input parameter gets its value from the uniform distribution. The set of adaptation parameters (here the total $CKR$ parameter, which is analyzed in the $CKR\_ACT$ and $CKR\_VCR$ parameters) is adjusted for each virtual patient, so that the actual medical outcome of the actual physical patient is optimally approximated in silico.

In order to properly assess the outcome of each execution of the clinical adaptation algorithm, it is essential to have good knowledge and understanding of the explored distribution for the values of the simulation input parameters, with respect to all the executed simulations, as well as with respect to the optimal adaptation simulations per virtual patient.

The probability distribution for each distinct simulation input parameter is visualized in the form of a normalized histogram. More specifically, the parameter values of the executed simulations (x axis) are matched to the number of their occurence divided by the total number of executed simulations (y axis), which corresponds to the probability with which each parameter value is represented in the simulation execution sample space. The normalized histograms that express the probability distribution of the input parameter values for each clinical adaptation execution of **Table 4.7** are depicted in **Figures 4.24, 4.25, 4.28, 4.29, 4.32, 4.33**.

The probability distribution for each parameter is visualized with respect to the sample space of all the simulation executions (red color), as well as with respect to the sample space that includes only the simulation executions that correspond to the optimal parameter adaptation per virtual patient (blue color). Since the clinical adaptation algorithm finds an optimal parameter adaptation for each virtual patient, the optimal adaptation sample space is a subspace of the overall execution sample space and the red and blue traces of the normalized histogram probability distribution figures mainly coincide.

The regions where the prevalent hue is blue correspond to the parameter values of the optimal adaptation simulations, which are thus considered to take part in a succesful patient representation in silico. This phenomenon is mainly observed for the set of adaptation parameters ($CKR\_ACT$ and $CKR\_VCR$) indicating the outcome of the clinical adaptation in the form of the desired distribution for the adaptation parameters that leads to the most fit representation of the patient in silico.

A variance between the red and blue hue histograms is also observed to a limited extent for the rest of the simulation parameters, indicating their corresponding values that were part of an optimal adaptation more frequently during the clinical adaptation execution. However, this variance is so minor that it does not lead to justifiable conclusions considering the optimal distribution for these parameters with respect to the optimal patient representation in silico.

The probability distribution for the parameters that do not belong to the set of adaptation parameters is displayed for the purpose of evaluating the degree in which the parameter values domain is properly represented in the clinical adaptation. Observing the normalized histogram figures and comparing their content for the varying number of virtual patients ($N = 20$ and $N = 200$), it becomes clear that as the number of virtual patients increases, the value domain of the uniform distribution is better represented.

The probability distribution of the values for each simulation parameter separately is not enough to fully describe the clinical adaptation algorithm execution. Each virtual patient of the clinical adaptation is characterized by a unique value for the joint distribution of the simulation parameters. However, the number of simulation parameters is too big for their joint distribution to be assessed effectively.

In order for the joint distribution for the pairs of a subset of selected simulation parameters to be evaluated, the pairplots of **Figures 4.26, 4.27, 4.30, 4.31, 4.34, 4.35** are provided. These plots express the probability distribution of the values for each parameter separately, as well as the joint distribution for each pair of parameters.

The pairplots consist of a series of subplots, which provide information concerning the probability distribution of the values for each parameter separately, as well as for the joint distribution of each pair of parameters. More specifically, each row and each column of subplots corresponds to a distinct parameter. The subplots that are located in the positions that correspond to row and column parameters that are the same (i.e. the subplots that are located in the diagonal) depict the probability distribution for the respective parameter as it is represented in the clinical adaptation execution, using the same methods as the ones that were employed for the construction of the normalized histograms. The probability distributions for the simulation parameters are evaluated within the sample space of all the simulation executions of the clinical adaptation (red trace), as well as within the sample space that contains the optimal with respect to the adaptation criterion simulations (blue trace).

The rest of the subplots are matched to positions that correspond to distinct row and column parameters and express the joint distribution of the respective pair of parameters in the form of a scatter plot of the cllinical adaptation execution simulation samples. The parameter that corresponds to the row position of the subplot defines the y coordinates of the simulation sample points in the scatter plot, while the parameter that corresponds to the column position of the subplot defines the x coordinates of the simulation sample points in the scatter plot. That way, the joint values that were represented in the clinical adaptation for the parameter pair are visualized. Moreover, the clinical adaptation error, with respect to the adaptation criterion (here the in silico tumor volume reduction and its deviation from the actual tumor volume reduction), is denoted by the color of the scatter points with reference to the depicted colorscale. The red scatter points correspond to simulation samples with high clinical adaptation error, while the blue scatter points correspond to simulation samples with low clinical adaptation error.

The subset of parameters for which the pairplots are provided includes the $Psym$ and $Psleep$ parameters, which express the rate of symmetric division of the cells that are in the mitotic phase and the rate with which the cells enter the dormant $G0$ phase respectively. These parameters have been proven, via the process of sensitivity analysis, to greatly affect the outcome of the simulation. Moreover, the adaptation parameters, $CKR\_VCR$ and $CKR\_ACT$, as well as the $Td$ parameter that expresses the doubling time of the tumor, are also included. The $Psym$, $Psleep$ and $Td$ parameters are assigned distinct joint values for each virtual patient. As a result, the number of simulation sample points of the pairplot scatter plots for these parameters is equal to the number of virtual patients N. On the other hand, a set of values is explored for the $CKR$ parameters for each virtual patient and the number of simulation sample points of the pairplot scatter plots for these parameters is equal to the total number of executed simulations within the clinical adaptation execution. The scatter plot between the $CKR\_VCR$ and the $CKR\_ACT$ parameters contains a limited number of point samples, since these parameters are expressed as a percentage of the total $CKR$ and subsequently are not independent.

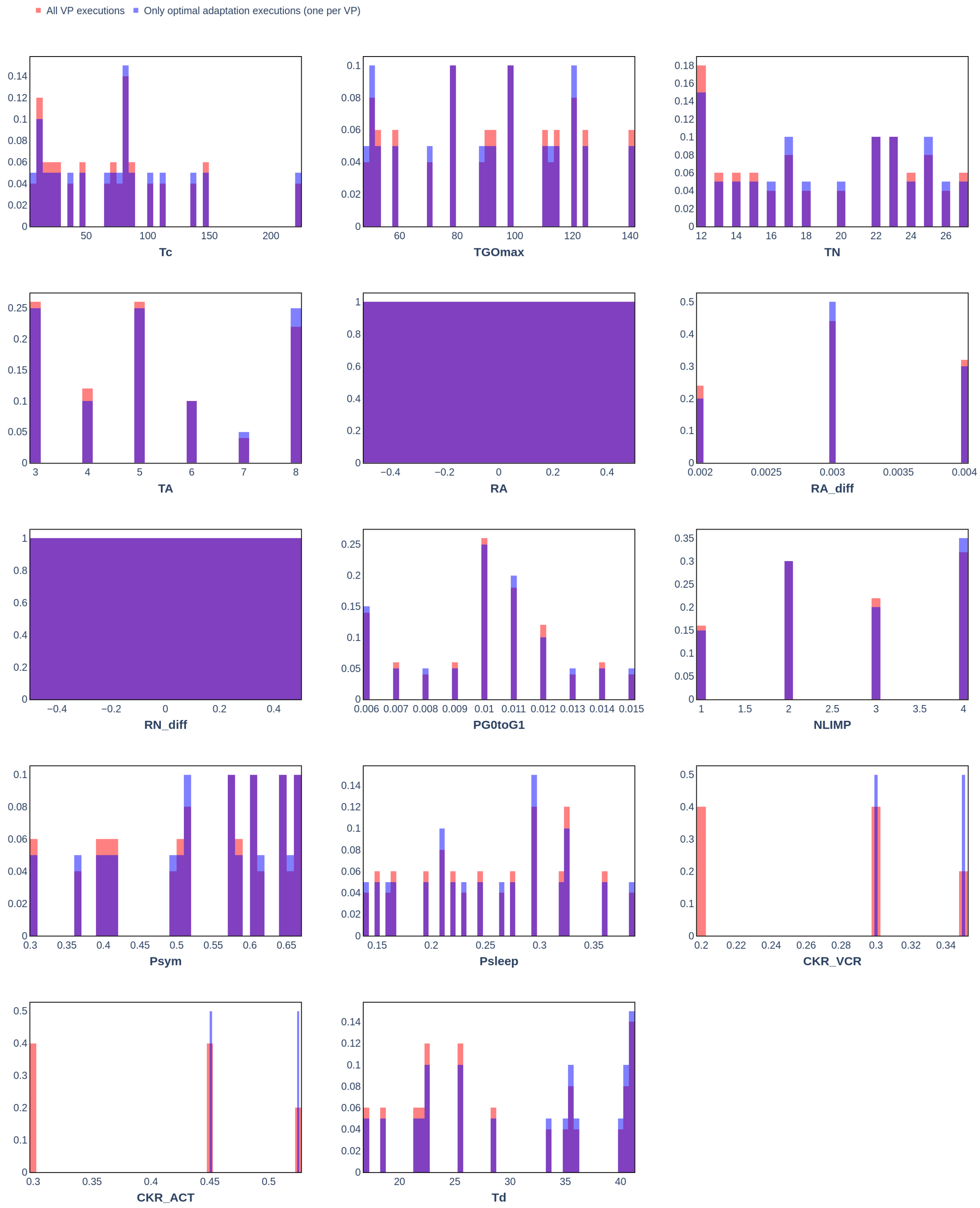


**Figure 4.24:** The normalized histograms that visualize the probability distribution of the simulation parameter values for the execution of the clinical adaptation algorithm for Patient_15 and number of virtual patients N=20. The red hue histograms refer to the sample space of all the executed simulations (see the eighth row of **Table 4.7** for the corresponding cardinality), while the blue hue histograms refer to the sample space that includes only the optimal adaptation simulations (one per VP, cardinality equal to $N = 20$). The optimal value range for each adaptation parameter is denoted by the regions of prevalence of the blue hue histogram as follows:
$CKR\_VCR = [0.3, 0.35]$ $CKR\_ACT = [0.45, 0.525]$ $CKR = [0.75, 0.875]$

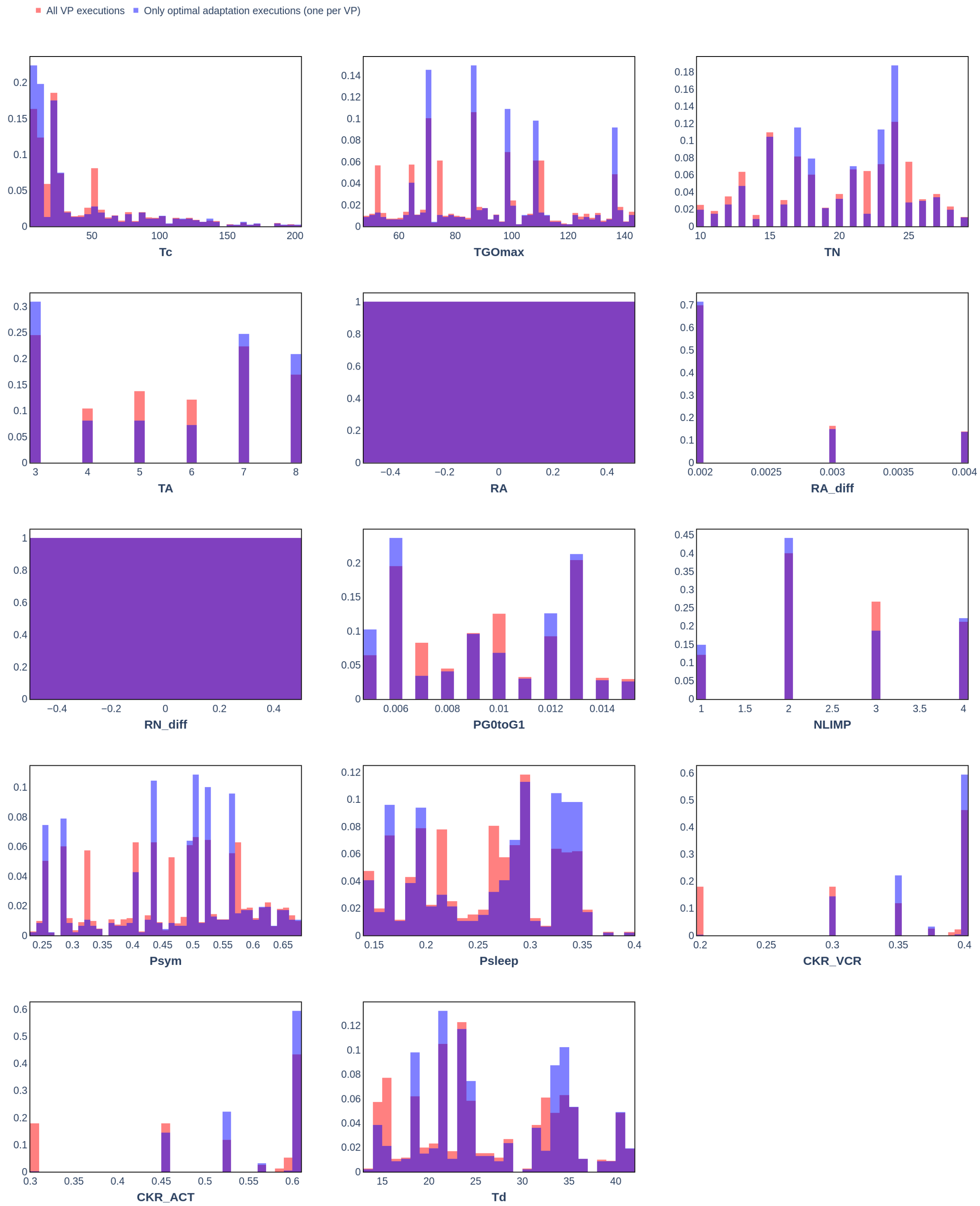


**Figure 4.25:** The normalized histograms that visualize the probability distribution of the simulation parameter values for the execution of the clinical adaptation algorithm for Patient_15 and number of virtual patients N=200. The red hue histograms refer to the sample space of all the executed simulations (see the eighth row of **Table 4.7** for the corresponding cardinality), while the blue hue histograms refer to the sample space that includes only the optimal adaptation simulations (one per VP, cardinality equal to $N = 200$). The optimal value range for each adaptation parameter is denoted by the regions of prevalence of the blue hue histogram as follows:

$CKR\_VCR = [0.3, 0.4]$ $\quad CKR\_ACT = [0.45, 0.6]$ $\quad CKR = [0.75, 1.0]$

Compared to the corresponding clinical adaptation execution for $N = 20$, the value range for the simulation parameters is better represented and the optimally adapted $CKR$ gets slightly higher values.

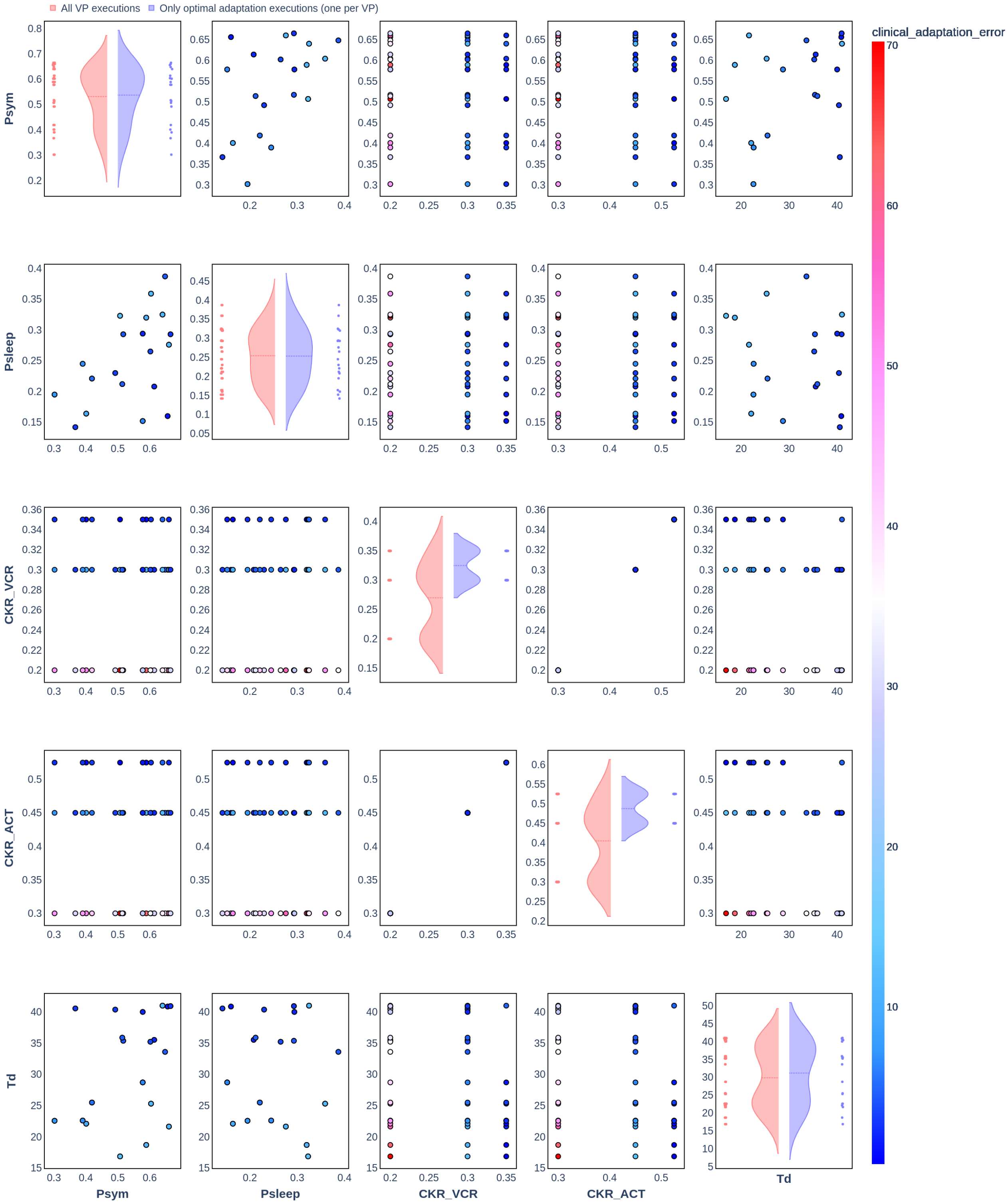


**Figure 4.26:** The pairplot that visualizes the joint distribution for the simulation parameter pairs for the execution of the clinical adaptation algorithm for Patient_15 and number of virtual patients N=20. The joint distribution for the pairs of the $Psym$, $Psleep$ and $Td$ parameters is random. The clinical adaptation error of the scatter point simulation samples is denoted by their color, with respect to the depicted colorscale. The distribution of the $CKR$ parameters is indicative of the exploration that was performed for their adjustment, while their joint distribution contains limited simulation sample scatter points, since their values are not independent. The mean values for the distribution of the $CKR$ and $Td$ parameters with respect to the optimal adaptation sample space (blue trace) are the following: $CKR\_VCR = 0.325 \qquad CKR\_ACT = 0.4875 \qquad CKR = 0.8125 \qquad Td = 31$

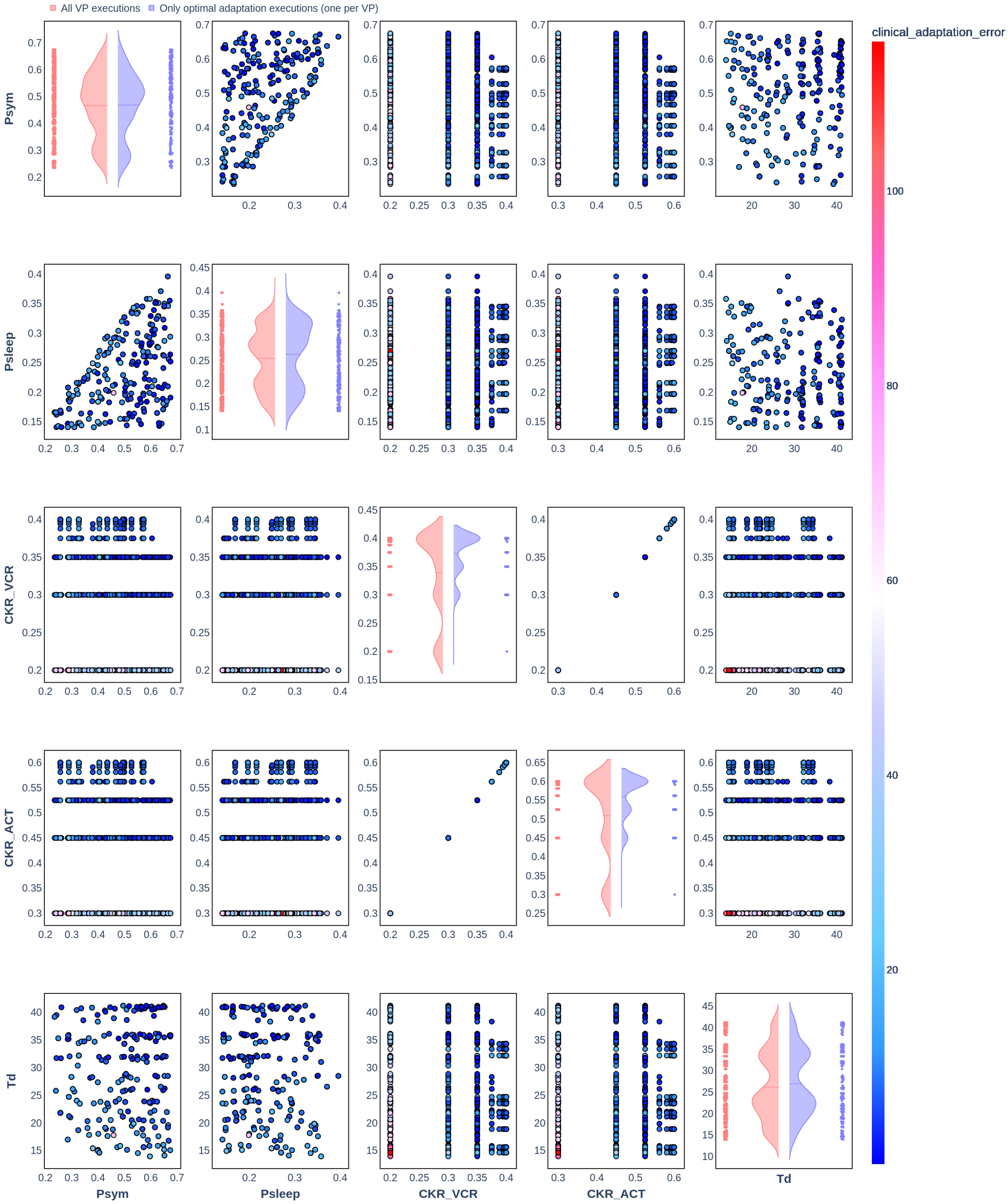


**Figure 4.27:** The pairplot that visualizes the joint distribution for the simulation parameter pairs for the execution of the clinical adaptation algorithm for Patient_15 and number of virtual patients N=200. The joint distribution for the pairs of the $Psym$, $Psleep$ and $Td$ parameters is random. The clinical adaptation error of the scatter point simulation samples is denoted by their color, with respect to the depicted colorscale. The distribution of the $CKR$ parameters is indicative of the exploration that was performed for their adjustment, while their joint distribution contains limited simulation sample scatter points, since their values are not independent. The mean values for the distribution of the $CKR$ and $Td$ parameters with respect to the optimal adaptation sample space (blue trace) are the following: $CKR\_VCR = 0.3732 \quad CKR\_ACT = 0.56 \quad CKR = 0.933 \quad Td = 27$

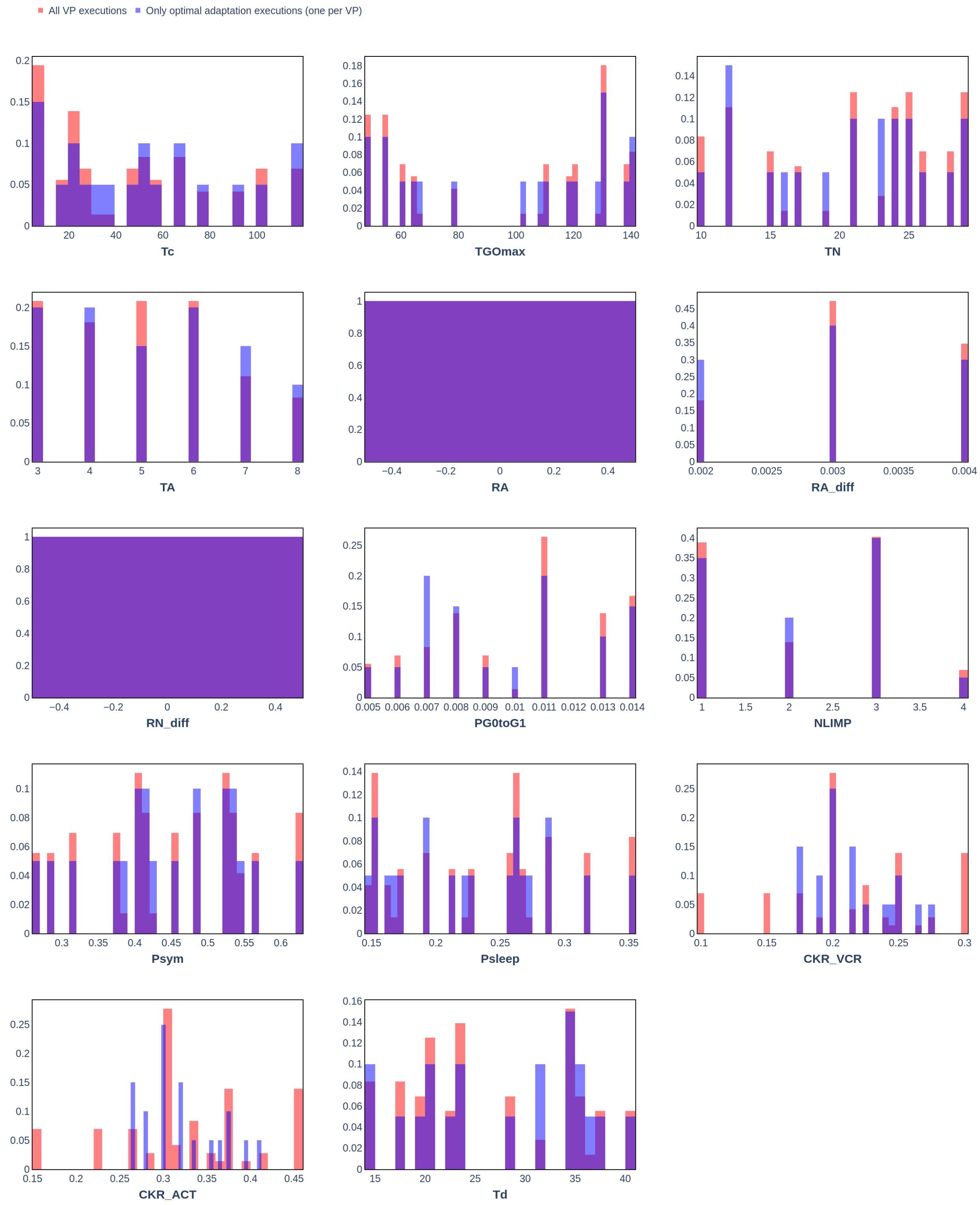


**Figure 4.28:** The normalized histograms that visualize the probability distribution of the simulation parameter values for the execution of the clinical adaptation algorithm for Patient_16 and number of virtual patients N=20. The red hue histograms refer to the sample space of all the executed simulations (see the eighth row of **Table 4.7** for the corresponding cardinality), while the blue hue histograms refer to the sample space that includes only the optimal adaptation simulations (one per VP, cardinality equal to $N = 20$). The optimal value range for each adaptation parameter is denoted by the regions of prevalence of the blue hue histogram as follows:

$CKR\_VCR = [0.175, 0.275]$ $CKR\_ACT = [0.263, 0.412]$ $CKR = [0.4375, 0.6875]$

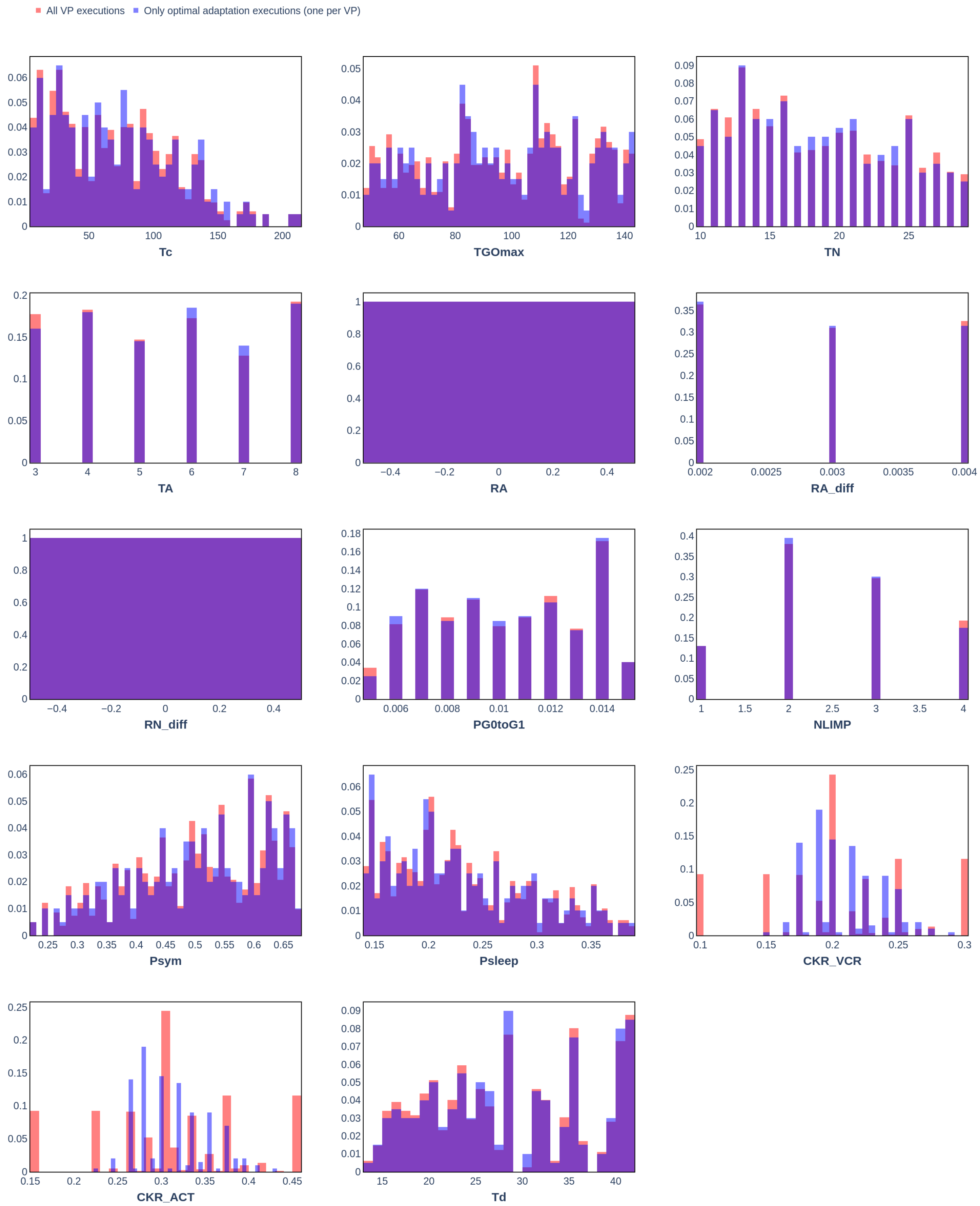


**Figure 4.29:** The normalized histograms that visualize the probability distribution of the simulation parameter values for the execution of the clinical adaptation algorithm for Patient_16 and number of virtual patients N=200. The red hue histograms refer to the sample space of all the executed simulations (see the eighth row of **Table 4.7** for the corresponding cardinality), while the blue hue histograms refer to the sample space that includes only the optimal adaptation simulations (one per VP, cardinality equal to $N = 200$). The optimal value range for each adaptation parameter is denoted by the regions of prevalence of the blue hue histogram as follows:
$CKR\_VCR = [0.15, 0.288]$ $CKR\_ACT = [0.225, 0.431]$ $CKR = [0.375, 0.72]$
Compared to the corresponding clinical adaptation execution for $N = 20$, the value range for the simulation parameters is better represented and the optimally adapted $CKR$ gets values from a slightly higher range.

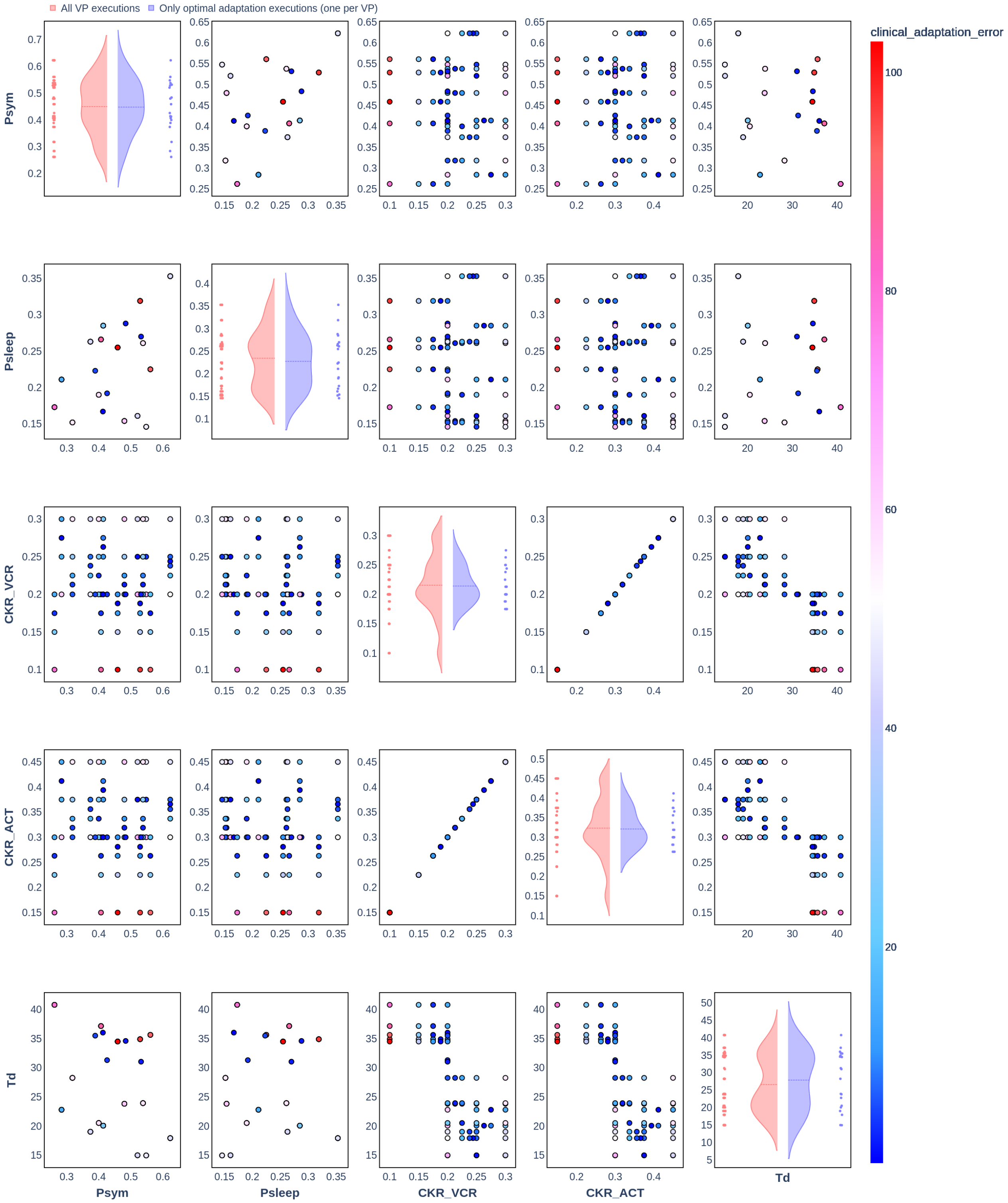


**Figure 4.30:** The pairplot that visualizes the joint distribution for the simulation parameter pairs for the execution of the clinical adaptation algorithm for Patient_16 and number of virtual patients N=20. The joint distribution for the pairs of the $Psym$, $Psleep$ and $Td$ parameters is random. The clinical adaptation error of the scatter point simulation samples is denoted by their color, with respect to the depicted colorscale. The distribution of the $CKR$ parameters is indicative of the exploration that was performed for their adjustment, while their joint distribution contains limited simulation sample scatter points, since their values are not independent. The mean values for the distribution of the $CKR$ and $Td$ parameters with respect to the optimal adaptation sample space (blue trace) are the following: $CKR\_VCR = 0.2143 \quad CKR\_ACT = 0.3212 \quad CKR = 0.5358 \quad Td = 28$

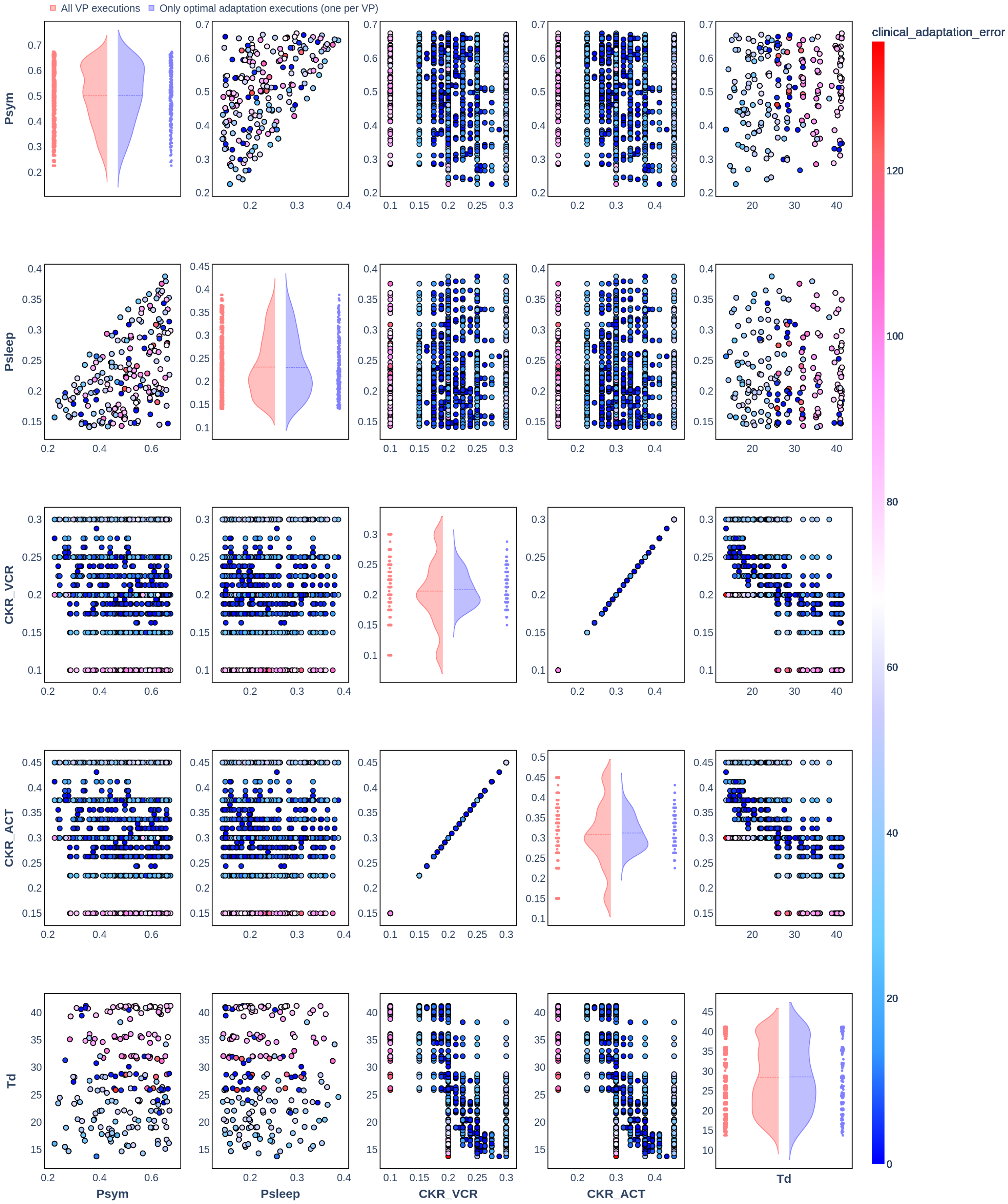


**Figure 4.31:** The pairplot that visualizes the joint distribution for the simulation parameter pairs for the execution of the clinical adaptation algorithm for Patient_16 and number of virtual patients N=200. The joint distribution for the pairs of the $Psym$, $Psleep$ and $Td$ parameters is random. The clinical adaptation error of the scatter point simulation samples is denoted by their color, with respect to the depicted colorscale. The distribution of the $CKR$ parameters is indicative of the exploration that was performed for their adjustment, while their joint distribution contains limited simulation sample scatter points, since their values are not independent. The mean values for the distribution of the $CKR$ and $Td$ parameters with respect to the optimal adaptation sample space (blue trace) are the following: $CKR\_VCR = 0.2084 \quad CKR\_ACT = 0.3123 \quad CKR = 0.521 \quad Td = 29$

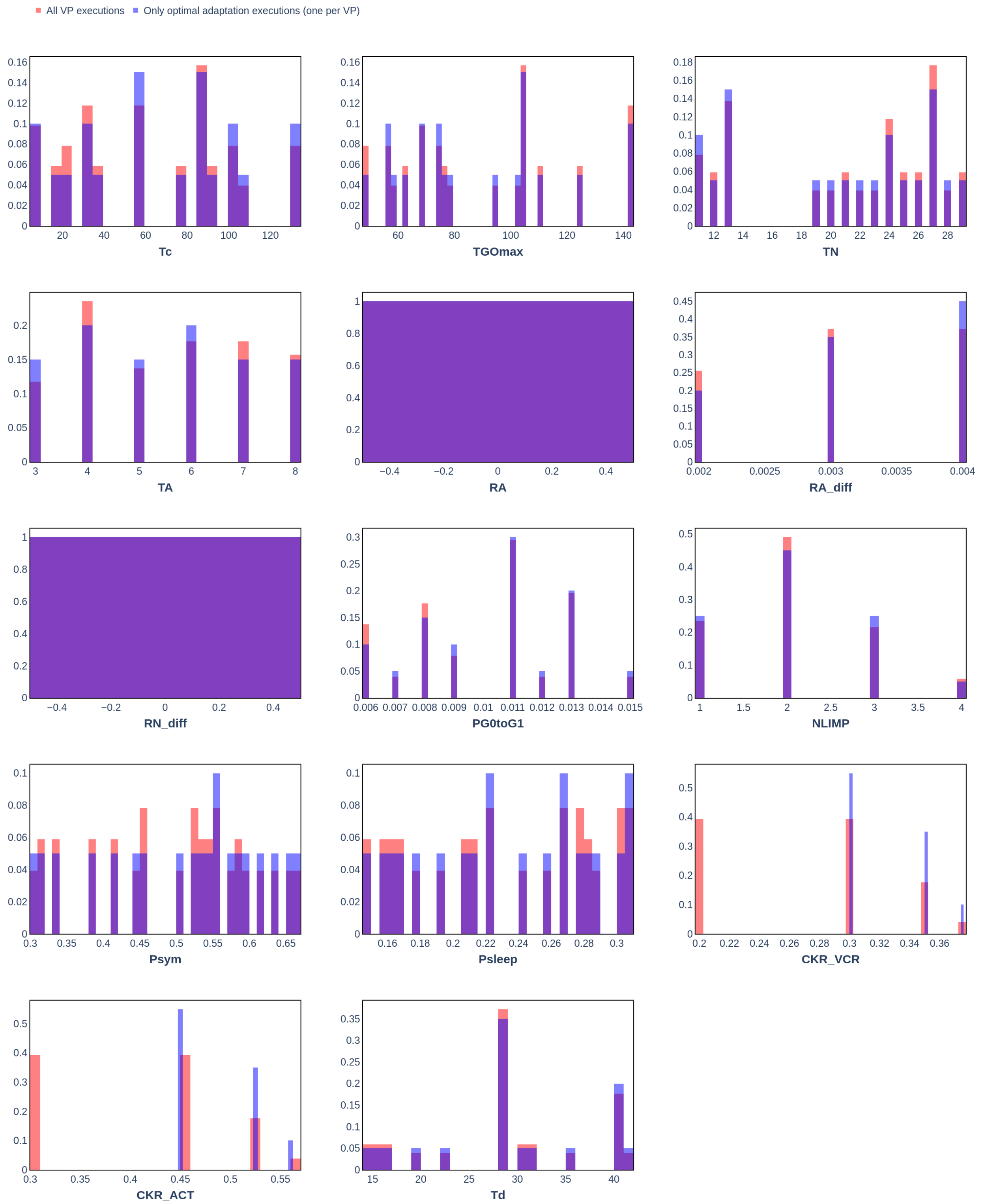


**Figure 4.32:** The normalized histograms that visualize the probability distribution of the simulation parameter values for the execution of the clinical adaptation algorithm for Patient_17 and number of virtual patients N=20. The red hue histograms refer to the sample space of all the executed simulations (see the eighth row of **Table 4.7** for the corresponding cardinality), while the blue hue histograms refer to the sample space that includes only the optimal adaptation simulations (one per VP, cardinality equal to $N = 20$). The optimal value range for each adaptation parameter is denoted by the regions of prevalence of the blue hue histogram as follows:
$CKR\_VCR = [0.3, 0.375]$ $CKR\_ACT = [0.45, 0.562]$ $CKR = [0.75, 0.9375]$

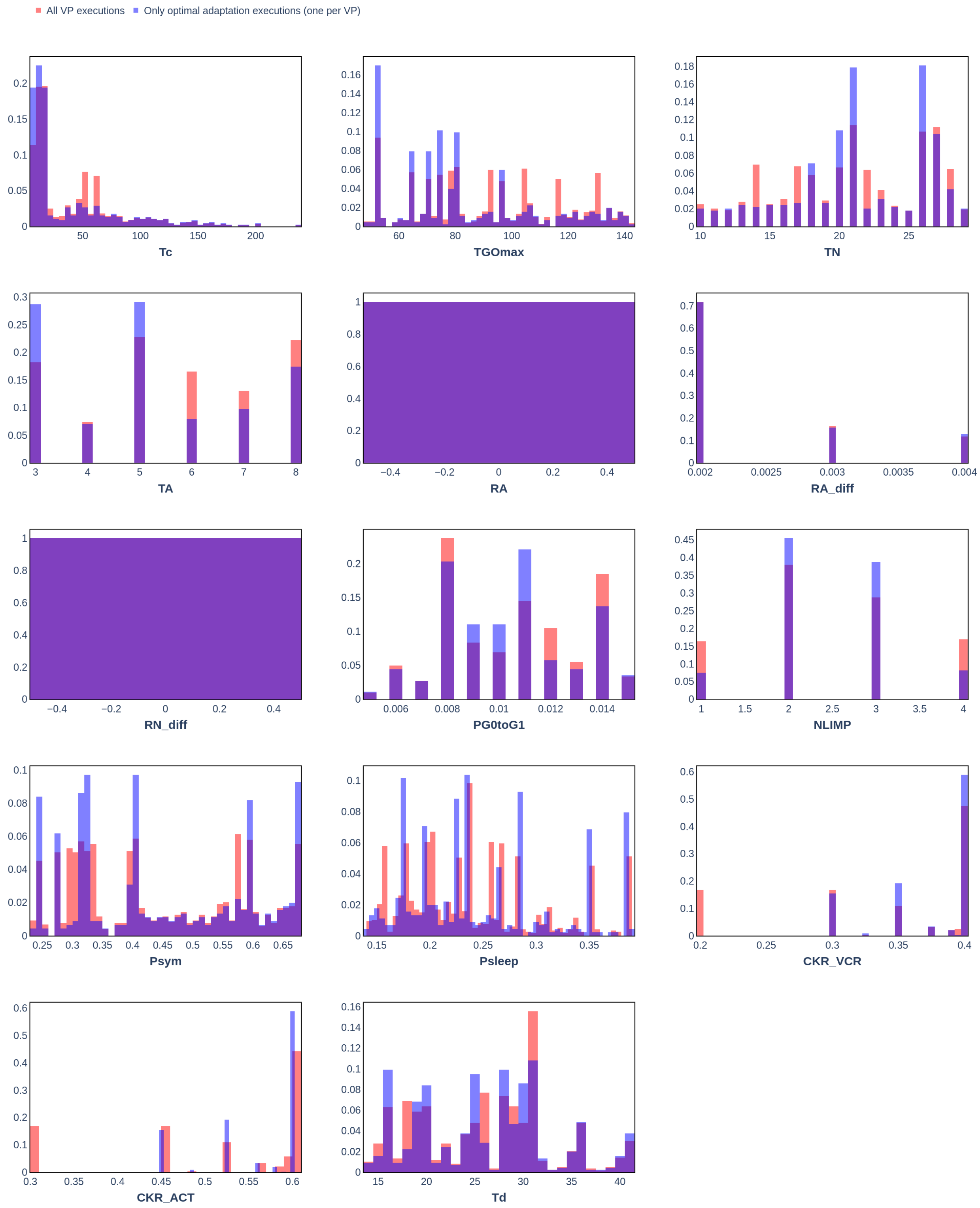


**Figure 4.33:** The normalized histograms that visualize the probability distribution of the simulation parameter values for the execution of the clinical adaptation algorithm for Patient_17 and number of virtual patients N=200. The red hue histograms refer to the sample space of all the executed simulations (see the eighth row of **Table 4.7** for the corresponding cardinality), while the blue hue histograms refer to the sample space that includes only the optimal adaptation simulations (one per VP, cardinality equal to $N = 200$). The optimal value range for each adaptation parameter is denoted by the regions of prevalence of the blue hue histogram as follows:
$CKR\_VCR = [0.3, 0.4]$ $\quad$ $CKR\_ACT = [0.45, 0.6]$ $\quad$ $CKR = [0.75, 1.0]$
Compared to the corresponding clinical adaptation execution for $N = 20$, the value range for the simulation parameters is better represented and the optimally adapted $CKR$ gets values from a slightly higher range.

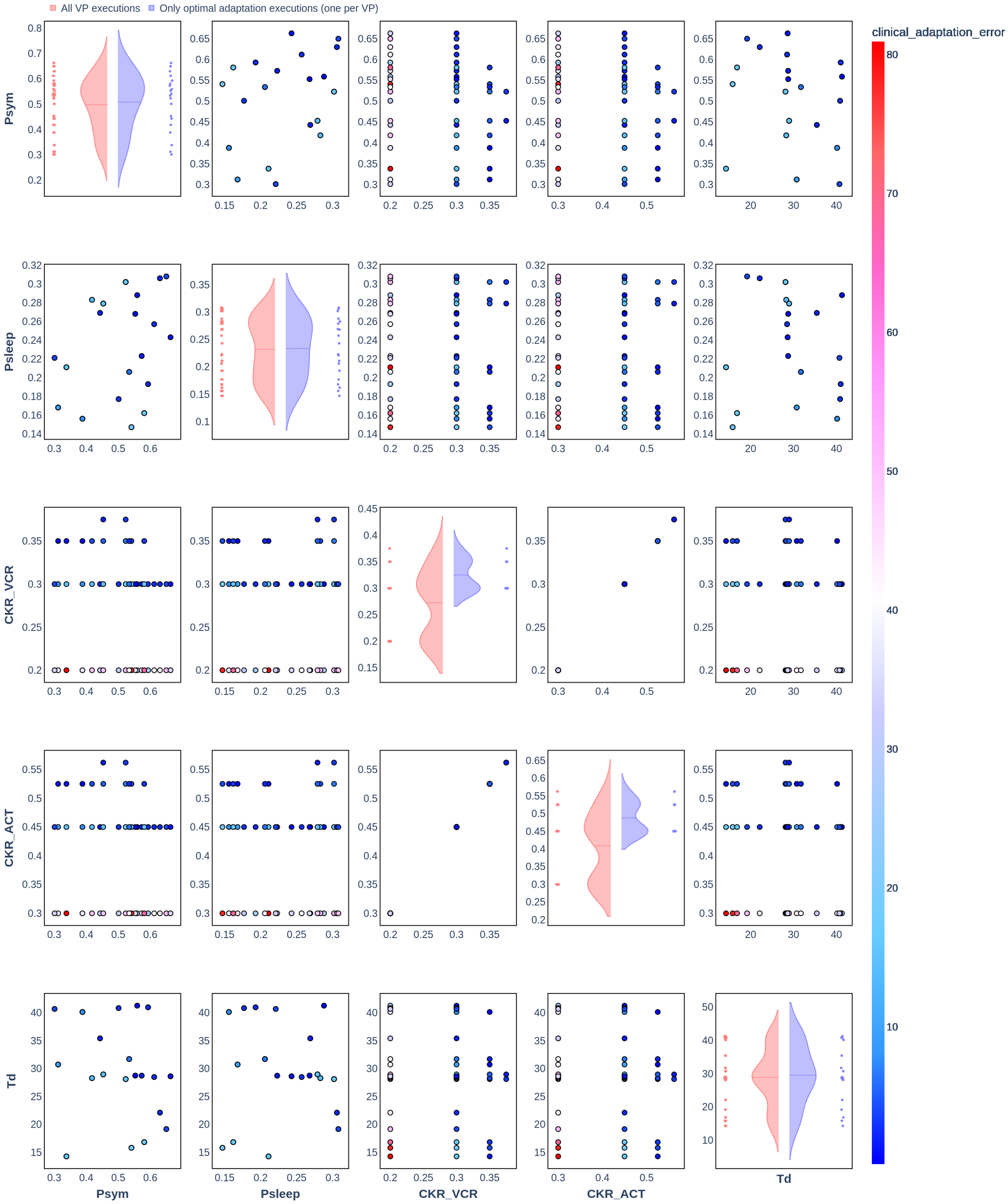


**Figure 4.34:** The pairplot that visualizes the joint distribution for the simulation parameter pairs for the execution of the clinical adaptation algorithm for Patient_17 and number of virtual patients N=20. The joint distribution for the pairs of the $Psym$, $Psleep$ and $Td$ parameters is random. The clinical adaptation error of the scatter point simulation samples is denoted by their color, with respect to the depicted colorscale. The distribution of the $CKR$ parameters is indicative of the exploration that was performed for their adjustment, while their joint distribution contains limited simulation sample scatter points, since their values are not independent. The mean values for the distribution of the $CKR$ and $Td$ parameters with respect to the optimal adaptation sample space (blue trace) are the following: $CKR\_VCR = 0.325 \quad CKR\_ACT = 0.4875 \quad CKR = 0.8125 \quad Td = 29$

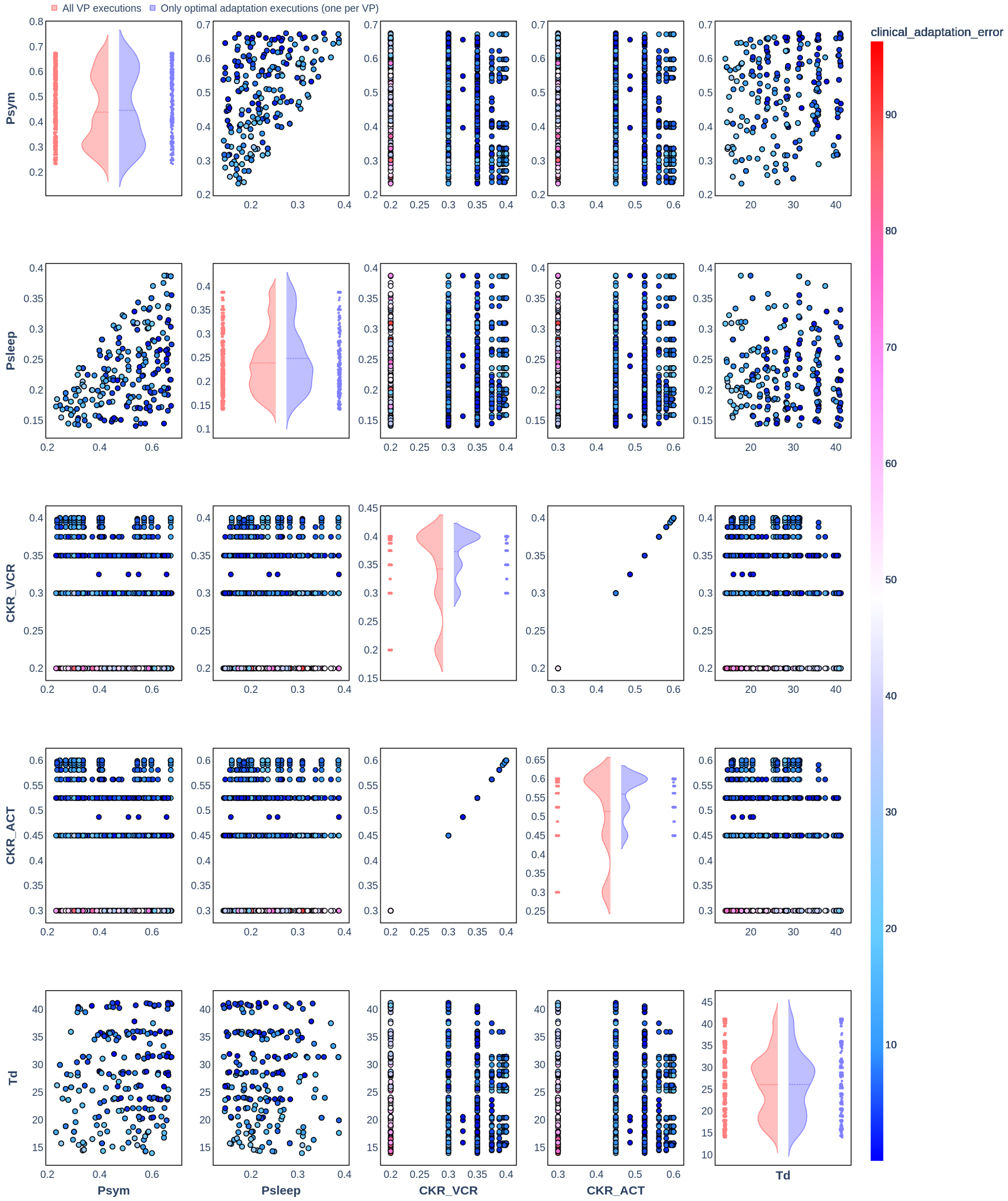


**Figure 4.35:** The pairplot that visualizes the joint distribution for the simulation parameter pairs for the execution of the clinical adaptation algorithm for Patient_17 and number of virtual patients N=200. The joint distribution for the pairs of the $Psym$, $Psleep$ and $Td$ parameters is random. The clinical adaptation error of the scatter point simulation samples is denoted by their color, with respect to the depicted colorscale. The distribution of the $CKR$ parameters is indicative of the exploration that was performed for their adjustment, while their joint distribution contains limited simulation sample scatter points, since their values are not independent. The mean values for the distribution of the $CKR$ and $Td$ parameters with respect to the optimal adaptation sample space (blue trace) are the following: $CKR\_VCR = 0.3732$ $\quad$ $CKR\_ACT = 0.5598$ $\quad$ $CKR = 0.933$ $\quad$ $Td = 26$

### 4.4.3 The evaluation of the clinical adaptation criterion for each virtual patient

The completion of the clinical adaptation algorithm requires the completion of the exploration of the set of adaptation parameters for each virtual patient. Here the set of adaptation parameters contains the total $CKR$ parameter, which is further analyzed into the $CKR\_VCR$ and the $CKR\_ACT$ components. The exploration of the $CKR$ parameter is performed for each virtual patient according to the algorithm of **Figure 3.10**. The value range that is explored is continuously adjusted, until a value that corresponds to a simulation with outcomes that satisfy the adaptation criterion is found.

Here the adaptation criterion refers to the in silico tumor volume reduction and the percentage by which it deviates from the actual tumor volume reduction. More specifically, the allowed deviation is limited to 5%. This means that for each virtual patient a value for the $CKR$ is determined ($N$ values in total), so that the in silico tumor volume reduction deviates from the actual tumor volume reduction by less than 5%.

The distribution of the optimal with respect to the clinical adaptation criterion $CKR$ values per virtual patient along the corresponding values for the in silico tumor volume reduction and the respective clinical adaptation error is displayed for each dataset patient and for the two distinct clinical adaptation executions with varying number of virtual patients ($N = 20$ and $N = 200$), in the histograms of **Figures 4.36-4.41**.

The displayed values for the CKR parameter are indeed the ones that are part of the optimal adaptation distribution that is visualized in the normalized histograms of the previous section. For the most part the adaptation criterion is satisfied for all the optimal adaptation values for the $CKR$ parameter. However, there exist some values where the clinical adaptation error threshold is transgressed. This indicates that for some virtual patients, their corresponding joint distribution value for the input parameters did not allow for the exploration of the $CKR$ parameter to find a value that approximates adequately in silico the actual tumor volume reduction. These cases should be noted so that they can be excluded from the final estimation of the $CKR$ distribution per dataset patient.

Moreover, even though they are not included in the adaptation criterion, the distribution along the in silico composition of the initial and the final tumor, as described by the percentages of the population for each cell type, are also displayed. These distributions can be taken into consideration when the risk profile of the patient, defined by the histologic profile and the macroscopic regressive changes, is known, as is the case here.

Patient_15 is characterized as low risk with a favorable monophasic histology and a high percentage of regressive changes, which means that an accurate in silico representation entails high percentages of differentiated and dead cells and limited percentages of dormant and especially proliferative cells.

Patient_16 is characterized as high risk with an unfavorable triphasic blastemal histology and a low percentage of regressive changes, which means that an accurate in silico representation entails low percentages of differentiated and dead cells and high percentages of dormant and especially proliferative cells.

Patient_17 is characterized as medium risk with a mixed biphasic histology and a medium percentage of regressive changes, which means that an accurate in silico representation entails mixed percentages for the various cell populations.

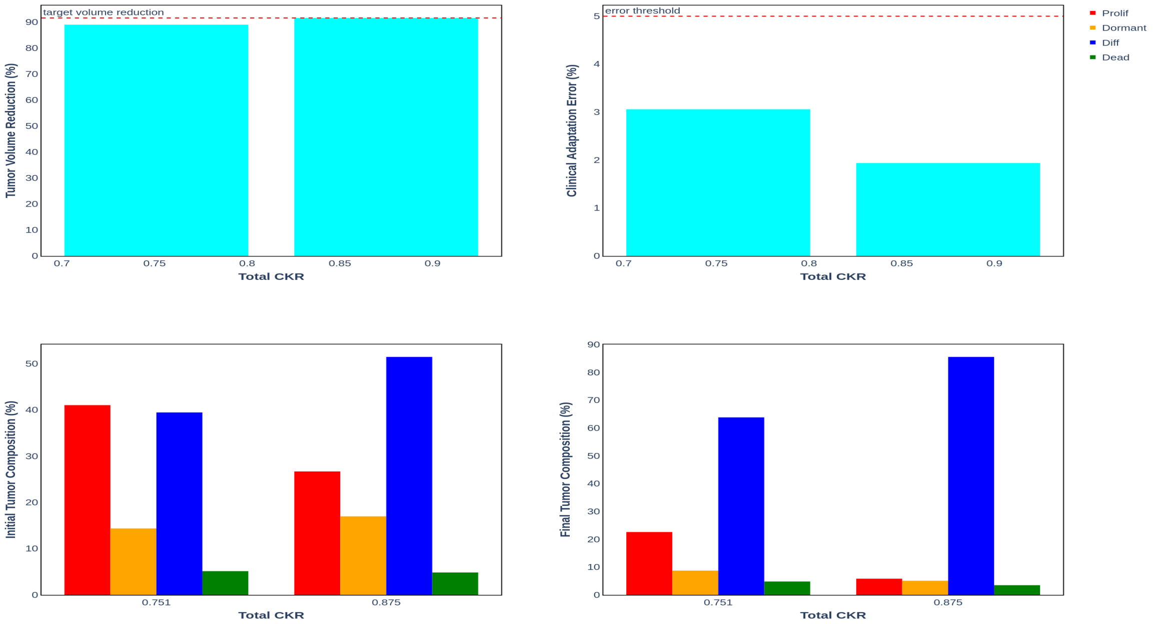


**Figure 4.36:** The histograms that visualize the relationship between the optimally adapted $CKR$ values and the in silico tumor volume reduction, the corresponding clinical adaptation error and the in silico composition of the initial and the final tumor for the execution of the clinical adaptation algorithm for Patient_15 and number of virtual patients N=20. The clinical adaptation criterion is not transgressed. Reviewing the in silico tumor composition, $CKR = 0.75$ is a more appropriate value for the in silico representation of the patient, since the corresponding histology is unfavorable.

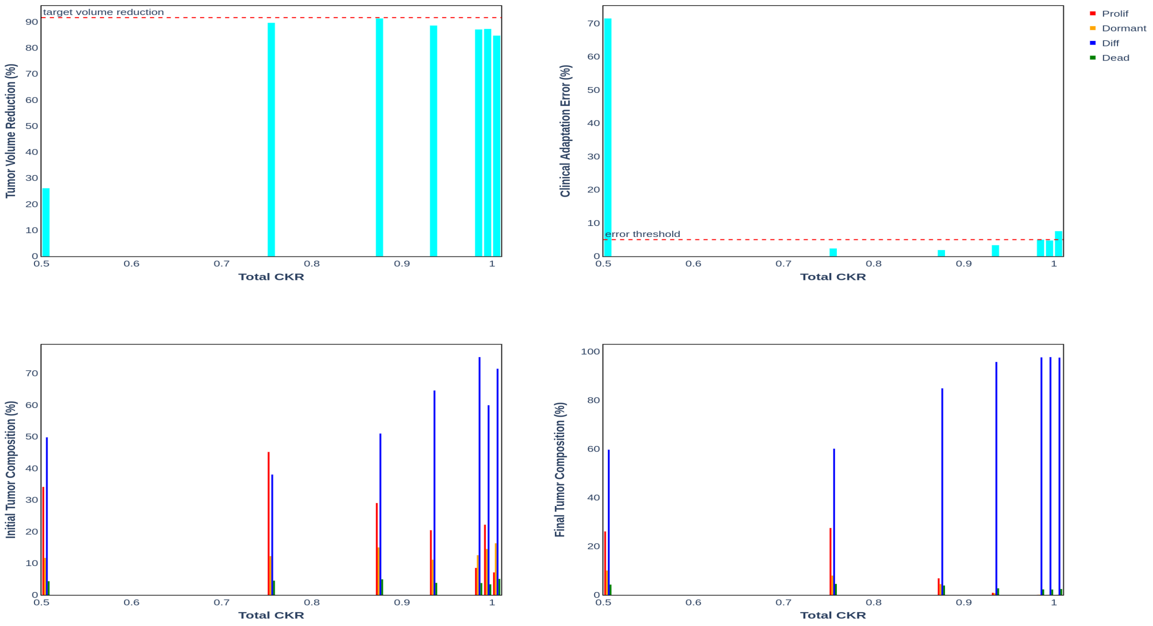


**Figure 4.37:** The histograms that visualize the relationship between the optimally adapted $CKR$ values and the in silico tumor volume reduction, the corresponding clinical adaptation error and the in silico composition of the initial and the final tumor for the execution of the clinical adaptation algorithm for Patient_15 and number of virtual patients N=200. The clinical adaptation criterion is transgressed for $CKR = 0.5$ and $CKR = 1.0$. Reviewing the in silico tumor composition, $CKR = 0.75$ is a more appropriate value for the in silico representation of the patient, since the corresponding histology is unfavorable.

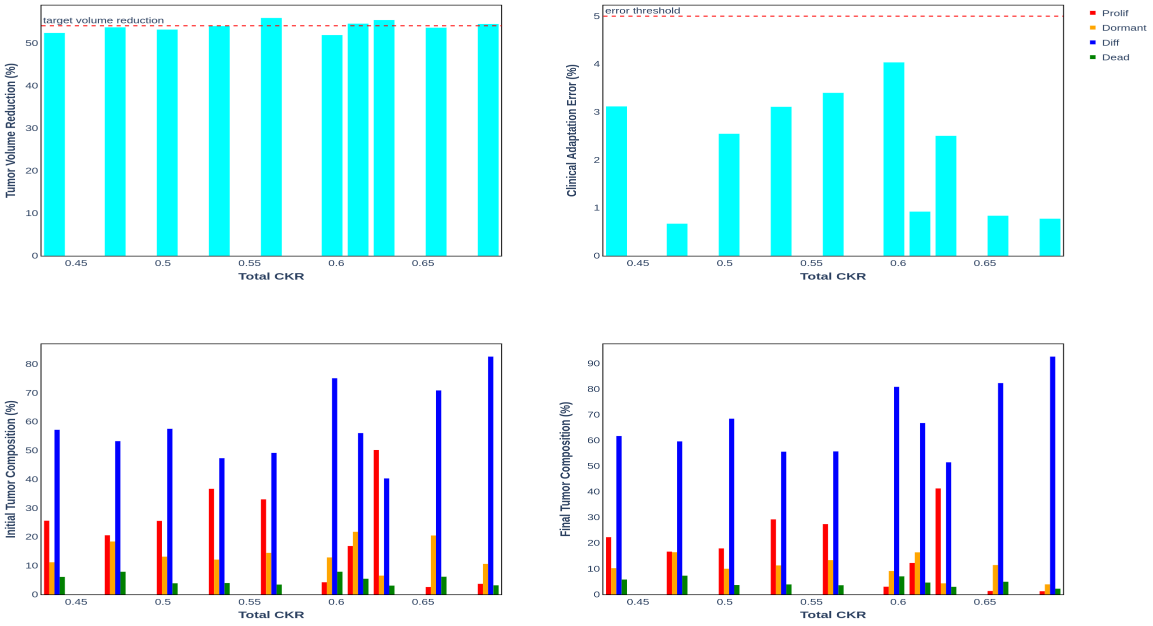


**Figure 4.38:** The histograms that visualize the relationship between the optimally adapted $CKR$ values and the in silico tumor volume reduction, the corresponding clinical adaptation error and the in silico composition of the initial and the final tumor for the execution of the clinical adaptation algorithm for Patient_16 and number of virtual patients N=20. The clinical adaptation criterion is not transgressed. Reviewing the in silico tumor composition, $CKR = 0.6 \bigcup [0.66, 0.69]$ is a more appropriate value range for the in silico representation of the patient, since the corresponding histology is favorable.

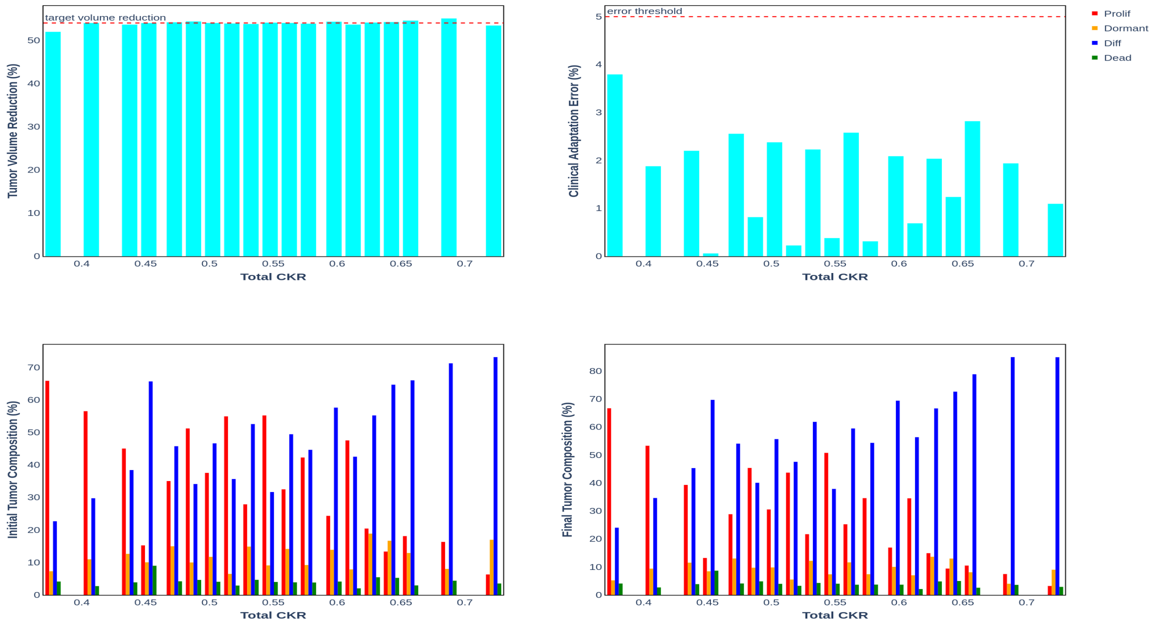


**Figure 4.39:** The histograms that visualize the relationship between the optimally adapted $CKR$ values and the in silico tumor volume reduction, the corresponding clinical adaptation error and the in silico composition of the initial and the final tumor for the execution of the clinical adaptation algorithm for Patient_16 and number of virtual patients N=200. The clinical adaptation criterion is not transgressed. Reviewing the in silico tumor composition, $CKR = 0.45 \bigcup 0.6 \bigcup [0.66, 0.69]$ is a more appropriate value range for the in silico representation of the patient, since the corresponding histology is favorable.

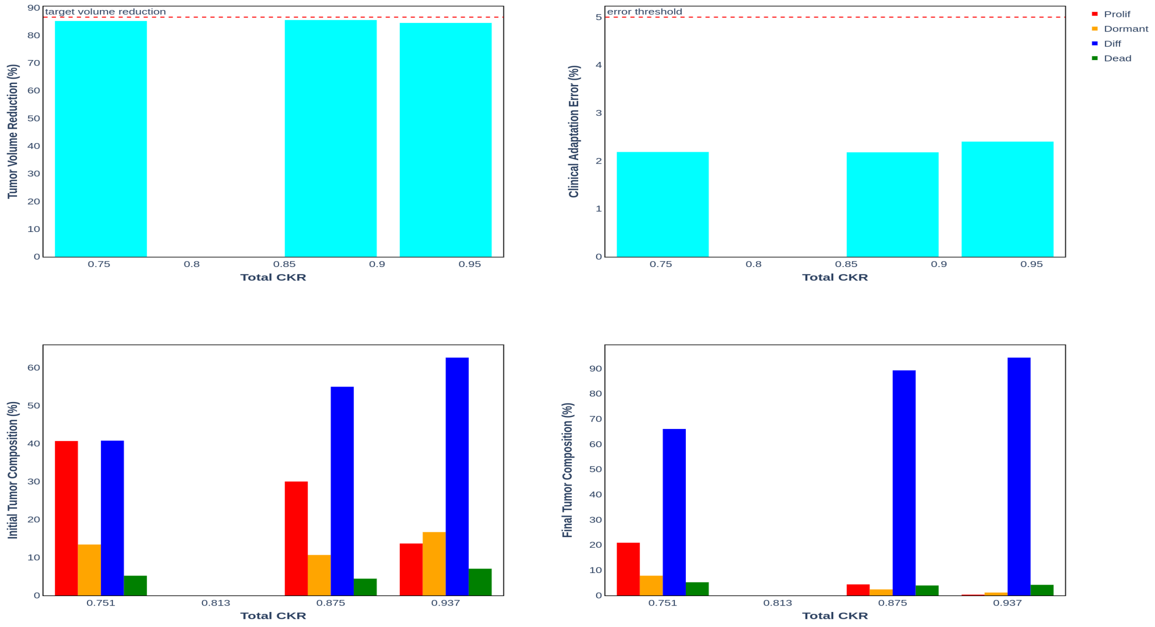


**Figure 4.40:** The histograms that visualize the relationship between the optimally adapted $CKR$ values and the in silico tumor volume reduction, the corresponding clinical adaptation error and the in silico composition of the initial and the final tumor for the execution of the clinical adaptation algorithm for Patient_17 and number of virtual patients N=20. The clinical adaptation criterion is not transgressed. Reviewing the in silico tumor composition, $CKR = 0.875$ is a more appropriate value for the in silico representation of the patient, since the corresponding histology is mixed.

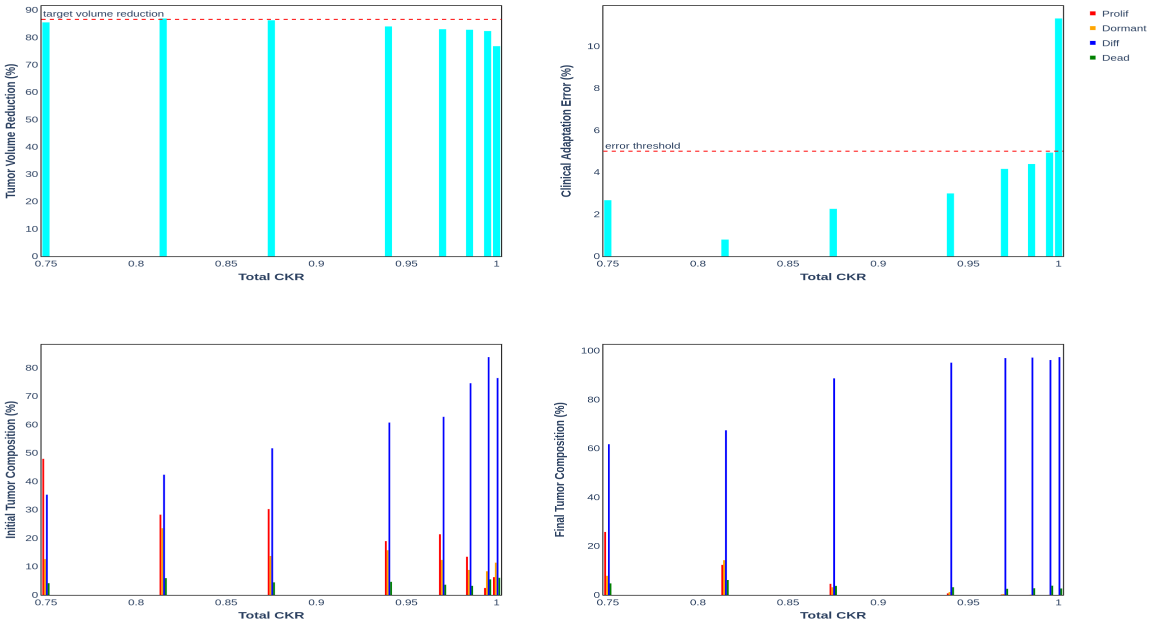


**Figure 4.41:** The histograms that visualize the relationship between the optimally adapted $CKR$ values and the in silico tumor volume reduction, the corresponding clinical adaptation error and the in silico composition of the initial and the final tumor for the execution of the clinical adaptation algorithm for Patient_17 and number of virtual patients N=200. The clinical adaptation criterion is transgressed for $CKR = 1.0$. Reviewing the in silico tumor composition, $CKR = [0.8125, 0.9724]$ is a more appropriate value range for the in silico representation of the patient, since the corresponding histology is mixed.

### 4.4.4 The resulting CKR distribution for each dataset patient

Reviewing the results of the clinical adaptation executions that were performed for each dataset patient (see **Table 4.7**), as they were presented in the previous sections with respect to the distributions for the values of the simulation input parameters both for the sample space of all the simulation executions and the sample space of the optimally adapted simulations per virtual patients, as well as with respect to the degree in which the adaptation criterion is satisfied for each virtual patient and the corresponding in silico tumor composition, the resulting distributions for the value of the $CKR$ parameter are presented in **Tables 4.8, 4.9 and 4.10** for each dataset patient.

The $CKR$ distributions are defined by the mean value and the standard deviation value of the execution samples that correspond to the optimally adapted simulations per virtual patient. Two versions of this set of values is provided, one considering only the tumor volume and one taking into consideration both the tumor volume and the tumor histology.

When only the tumor volume in considered, the number of samples that define the $CKR$ distribution are approximately equal to the number of virtual patients $N$, since each virtual patient explores the $CKR$ parameter values until a suitable value for the in silico representation of the patient is found. However, in some cases the valid samples are slightly lower than the number of virtual patients $N$, especially as $N$ increases, since not all distinct values for the joint distribution of the rest of the simulation parameters that the virtual patients represent are adequate for the simulation of the actual patient, no matter the value for the $CKR$ parameter.

When both the tumor volume and the tumor histology are taken into account, the number of samples that define the $CKR$ distribution drop dramatically, since only the virtual patient optimally adapted simulations that match the histology of the corresponding actual patient are considered. However, the tumor histology is not part of the adaptation criterion of the implemented clinical adaptation algorithm. As a result, only a percentage of the virtual patients with adapted values for the $CKR$ parameter are included in the distribution.

| | Number of samples | | Mean value for the $CKR$ | | Std value for the $CKR$ | |
|---|---|---|---|---|---|---|
| | $N = 20$ | $N = 200$ | $N = 20$ | $N = 200$ | $N = 20$ | $N = 200$ |
| **Only the tumor volume is considered** | 20 | 189 | 0.8125 | 0.83619 | 0.064124 | 0.067726 |
| **Both the tumor volume and the tumor histology are considered** | 3 | 46 | 0.875 | 0.87875 | 0 | 0.054272 |

**Table 4.8:** The resulting distribution of the $CKR$ parameter for Patient_15, as it is defined by the set of corresponding mean and standard deviation values that are represented in the sample space of the simulations that were executed during the clinical adaptation. When only the tumor volume is considered, the number of samples is approximating the number of virtual patients that are matched to the actual patient, since the clinical adaptation algorithm is designed to find a suitable value for the $CKR$ parameter per virtual patient. However, in some cases the value of the joint distribution for the rest of the simulation input parameters that a virtual patient represents is not suitable for the simulation of the actual patient with minimal error, no matter the value for the $CKR$ parameter. That is why the number of samples that define the $CKR$ distribution may be slightly lower that the number of virtual patients $N$ (here 189 instead of 200, see also **Figure 4.37**). When both the tumor volume and the tumor histology are considered, the number of samples drops even more, since the tumor histology is not part of the adaptation criterion and as a result only a percentage of adapted virtual patients defines the distribution. Here, only the virtual patient simulations that correspond to an in silico tumor that is comprised by differentiated cells by at least 60% were considered, since the patient histology is favorable, reducing the number of samples to 3 for $N = 20$ and to 46 for $N = 200$. When only the tumor volume is considered, the value for the $CKR$ parameter is slightly overestimated. In any case, the mean value is safely assumed to lie in the following window: $CKR_{MEAN} \in [0.81, 0.88]$.

| | Number of samples | | Mean value for the $CKR$ | | Std value for the $CKR$ | |
|---|---|---|---|---|---|---|
| | $N = 20$ | $N = 200$ | $N = 20$ | $N = 200$ | $N = 20$ | $N = 200$ |
| **Only the tumor volume is considered** | 20 | 200 | 0.535625 | 0.521037 | 0.075236 | 0.067967 |
| **Both the tumor volume and the tumor histology are considered** | 4 | 77 | 0.555 | 0.48526 | 0.089652 | 0.054546 |

**Table 4.9:** The resulting distribution of the $CKR$ parameter for Patient_16, as it is defined by the set of corresponding mean and standard deviation values that are represented in the sample space of the simulations that were executed during the clinical adaptation. When only the tumor volume is considered, the number of samples is equal to the number of virtual patients, which indicates that all the values of the joint distribution for the rest of the simulation input parameters that the virtual patients represent are suitable for the simulation of the actual patient with minimal error, when the $CKR$ parameter value is adjusted appropriately. When both the tumor volume and the tumor histology are considered, the number of samples drops, since, in order to represent the unfavorable histology of the actual patient, only the virtual patient simulations that correspond to an in silico tumor that is comprised by proliferative cells by at least 40% were considered, reducing the number of samples to 4 for $N = 20$ and to 77 for $N = 200$. When only the tumor volume is considered, the value for the $CKR$ parameter is slightly underestimated. In any case, the mean value is safely assumed to lie in the following window: $CKR_{MEAN} \in [0.49, 0.56]$.

|  | Number of samples |  | Mean value for the $CKR$ |  | Std value for the $CKR$ |  |
|---|---|---|---|---|---|---|
|  | $N = 20$ | $N = 200$ | $N = 20$ | $N = 200$ | $N = 20$ | $N = 200$ |
| **Only the tumor volume is considered** | 20 | 188 | 0.8125 | 0.838537 | 0.073112 | 0.074602 |
| **Both the tumor volume and the tumor histology are considered** | 5 | 76 | 0.8625 | 0.86477 | 0.068465 | 0.06589 |

**Table 4.10:** The resulting distribution of the $CKR$ parameter for Patient_17, as it is defined by the set of corresponding mean and standard deviation values that are represented in the sample space of the simulations that were executed during the clinical adaptation. When only the tumor volume is considered, the number of samples is approximating the number of virtual patients that are matched to the actual patient, since the clinical adaptation algorithm is designed to find a suitable value for the $CKR$ parameter per virtual patient. However, in some cases the value of the joint distribution for the rest of the simulation input parameters that a virtual patient represents is not suitable for the simulation of the actual patient with minimal error, no matter the value for the $CKR$ parameter. That is why the number of samples that define the $CKR$ distribution may be slightly lower that the number of virtual patients $N$ (here 188 instead of 200, see also **Figure 4.41**). When both the tumor volume and the tumor histology are considered, the number of samples drops, since, in order to represent the mixed histology of the actual patient, only the virtual patient simulations that correspond to an in silico tumor that is composed by a percentage of differentiated cells that lies between 45% and 65% were considered, reducing the number of samples to 5 for $N = 20$ and to 76 for $N = 200$. When only the tumor volume is considered, the value for the $CKR$ parameter is slightly underestimated. In any case, the mean value is safely assumed to lie in the following window: $CKR_{MEAN} \in [0.81, 0.86]$.

# Conclusion & Future Work

The Nephroblastoma Oncosimulator is a discrete top-down model for multiscale cancer modeling that uses the patient-specific medical imaging data for the in silico representation of Wilms' tumor, which is the most common renal tumor in childhood, as a 3D matrix of voxels and the simulation of its temporal evolution in discrete time steps. The goal of the Nephroblastoma Oncosimulator is to predict the disease progression for a nephroblastoma patient and ultimately act as a medical advisor that offers insight on the best course of treatment for the specific patient. In order for that to be possible, the patient-specific medical data need to be translated in silico. However, the values for the simulation input parameters are not defined unambiguously by the patient-specific medical data, which renders the execution of multiple virtual patient simulations necessary for the adequate in silico representation of the physical patient. More specifically, each virtual patient is assigned a distinct value for the joint distribution of the simulation input parameters and the simulation outcomes are statistically coevaluated for the optimal in silico representation of the actual physical patient.

In order for the Oncosimulator to be used in clinical practice as a medical advisor application, it needs to meet the specifications of an on-line model that produces results in real time. In the present work, high performance computing methods were explored, for the optimization of the execution of a single simulation, as well as for the efficient execution of multiple virtual patient simulations. The performance optimization was achieved via the preprocessing of the simulation inputs and via source code modifications. That way, an optimized version of the model is emerged, whose functionality was verified via sensitivity analysis. After its verification, the optimized Oncosimulator model was utilized for the clinical adaptation of the simulation input parameters via the execution of multiple virtual patient simulations. More specifically, the CKR parameter, which defines the ratio of cells that are hit by therapy and thus expresses the treatment effectiveness, was adapted for each one of the three dataset patients that represent distinct histologic profiles and corresponding risk groups. That way, the probability distribution of the CKR parameter is defined for each disctinct patient group that is represented in the dataset.

The simulation execution profiling, which was perfomed using the valgrind framework, provided an estimate of the percentage with which each simulation scan contributes to the total cost of the simulation. Before the data preprocessing step, the third simulation scan, which implements the simulation on the cellular biocomplexity scale, takes up $0.6\% - 4\%$ of the total simulation execution time, while after data preprocessing is performed, it takes up $13\% - 20\%$ of the total simulation execution time. Moreover, the data preprocessing causes a dicrease of the memory footprint of the simulation execution by approximately 90%, as well as a dicrease of the total simulation execution time by more than 97%.

The OpenMP API was utilized for the modification of the source code, so that the performance of the computations is done concurrently and in parallel when the data dependencies allow it, i.e. for the third scan and the simulation on the cellular level of biocomplexity. All in all, the data preprocessing step, which adjusts the resolution and the cost of the simulation via the adjustment of the size of the voxel in the 3D matrix that represents the tumor in silico, reduces the simulation input size, introduces the greatest speedup for the simulation execution time and at the same time alters the workload balance between the simulation scans, rendering it more fair and less intense on the parts that do not benefit from the optimized resources, i.e. the multiple threads, and indirectly improving the speedup potential of the code optimization step according to Amdahl's law. Finally, after data preprocessing is performed, the code optimization step that accelerates the third simulation scan introduces a speedup of approximately $16\% - 24\%$, which is still minor compared to the speedup that is introduced by the data preprocessing step.

Before the clinical adaptation is performed, the functionality of the Oncosimulator optimized version, as it is derived after the data preprocessing and the code optimization modifications, is verified via sensitivity analysis, i.e. via the exploration of the simulation input parameter values and the manner in which they affect the simulation outcomes. Indeed, the model behavior is the expected one. For instance, as the $CKR$ parameter is increased so is the tumor volume reduction, as the $P_{sym}$ parameter is increased the tumor growth is more aggressive, while as the $P_{sleep}$ parameter is increased the tumor growth is inhibited. The clinical adaptation step is performed using the new optimized and verified model for the execution of multiple virtual patient simulations per physical patient that represents a distinct histologic and risk group in the dataset. That way, the probability distribution of the CKR parameter for each distinct patient group of the dataset is derived as a set of mean and standard deviation values. The low risk group corresponds to a mean value of 0.9 for the CKR parameter, the high risk group corresponds to a mean value of 0.5, while the medium risk group corresponds to a mean value of 0.8.

The following steps to be implemented in the future work can focus on the refinement and enrichment of the algorithms and methods that are utilized in the already implemented steps, as well as on taking further action toward the utilization of the Nephroblastoma Oncosimulator in clinical practice. With concern to the efficient execution of a single simulation, the performed profiling records the simulation scans on the tissue biocomplexity level as the most costly part of the simulation, taking up to 80% of the total simulation execution time, even after the data preprocessing step is performed. The bottleneck that this part of the simulation introduces needs to be addressed by altering the source code, so that the data dependencies are eliminated as much as possible. Moreover, the source code can be enriched so that the tumor histology is also represented in the 3D tumor matrix. In that context, the adjustment of the geometrical cell size that addresses the simulation resolution and cost tradeoff can be performed dynamically during the simulation execution, depending on the computational needs and separately for each histologic region of the 3D tumor matrix. In any case, the source code modifications should be done so that the data dependencies are kept limited, the workload is fairly balanced between the simulation scans and the conditions for the productive utilization of mutlithreading according to Amdahl's law are retained.

After the source code is modified and prepared for the code optimization step, the performance goals for a single simulation execution can be set and an optimization plan can be devised and implemented, in order to better accommodate these goals. More specifically, the performance goals for a single simulation execution depend on the performance standards that are required by the clinical adaptation step that uses multiple simulation executions for the in silico representation of the physical patient using multiple virtual patients. In order to evaluate the performance goals for a single simulation execution, the parameters of the clinical adaptation algorithm, e.g. the number of virtual patients N, need to be explored and their optimal values determined. Then, the code optimization can take place and the architecture that better accommodates the performance goals can be implemented. This architecture can be heterogeneous. For instance, the third simulation scan includes independent computations for the voxels of the 3D tumor matrix, that can be performed concurrently and in parallel using a GPU architecture.

Then, the architecture for the execution of multiple virtual patient simulations per physical patient can be implemented and utilized for the determination of the probability distribution of the simulation input parameters, the same way that was done in the present work for the CKR parameter. For that purpose, the enrichment of the patient dataset is crucial, so that more histologic and risk groups are represented in it and with more samples each. After the distribution of the simulation parameters is determined for the distinct patient groups of the dataset, the multiple virtual patient simulation executions can be utilized for the in silico representation of patients that have not yet been treated, in order to predict their clinical outcomes and suggest the best treatment course. More specifically, after the patient group of the dataset to which the current patient belongs is determined, multiple virtual patient executions are performed, with each virtual patient being assigned values for the simulation input parameters according to the parameter distributions for the specific patient group, and the simulation outcomes are statistically coevaluated for the prediction of the actual clinical outcomes of the physical patient. That way, the Nephroblastoma Oncosimulator can start to be used and tested in clinical practice and clinical trials, delivered as an integrated application, with a proper DBMS and user interface.

# Bibliography

[1] C. Sonnenschein and A. Soto, "The aging of the 2000 and 2011 hallmarks of cancer reviews: A critique," *Journal of biosciences*, vol. 38, pp. 651–63, 09 2013. Retrieved 20 January 2021 from https://www.cell.com/fulltext/S0092-8674(11)00127-9.

[2] T. S. Deisboeck, Z. Wang, P. Macklin, and V. Cristini, "Multiscale cancer modeling," *Annual review of biomedical engineering*, vol. 13, pp. 127–155, Aug 2011.

[3] T. S. Deisboeck, Z. Wang, P. Macklin, and V. Cristini, "Multiscale cancer modeling," *Annual Review of Biomedical Engineering*, vol. 13, no. 1, pp. 127–155, 2011. PMID: 21529163.

[4] Retrieved 20 January 2021 from http://mycourses.ntua.gr/courses/ECE1406/document/01_%D0%D0%CA%C9S%C9_-705-_%CC%C5%D4%C1%D0%D4_%D3%C7%CC%CC%D5_%C5%CC%D0_%CF%CB%C5%D3_%CF%C9_%C4%C9%C1%CB%C5%CE%C5%C9%D3_%D4%C5%D4%C1%D1%D4%C7_%C5%CA%C4%CF%D3%C7_%C3%D3%D3_2020_NOE_20_%D4_18_53_EST.pdf.

[5] S. Wan and P. V. Coveney, "Molecular dynamics simulation reveals structural and thermodynamic features of kinase activation by cancer mutations within the epidermal growth factor receptor," *Journal of Computational Chemistry*, vol. 32, no. 13, pp. 2843–2852, 2011.

[6] P. Castorina, T. Deisboeck, P. Gabriele, and C. Guiot, "Growth laws in cancer: Implications for radiotherapy," *Radiation research*, vol. 168, pp. 349–56, 10 2007.

[7] G. Stamatakos, D. Dionysiou, S. Giatili, E. Kolokotroni, E. Georgiadi, A. Roniotis, V. Sakkalis, S. Wan, S. Manos, S. Zasada, A. Folarin, P. Büchler, T. Bardyn, S. Bauer, M. Reyes, T. Bily, V. Bednar, M. Karásek, N. Graf, and K. Marias, "The contracancrum oncosimulator: Integrating biomechanisms across scales in the clinical context," 09 2010.

[8] A. Roniotis, K. Panourgias, J. Ekaterinaris, K. Marias, and V. Sakkalis, "Approximating the diffusion–reaction equation for developing glioma models for the contracancrum project: a showcase," 09 2010.

[9] S. Giatili, N. Uzunoglu, and G. Stamatakos, "An explicit boundary condition treatment of a diffusion – based glioblastoma tumor growth model," 09 2010.

[10] S. Giatili and G. Stamatakos, "A detailed numerical treatment of the boundary conditions imposed by the skull on a diffusion–reaction model of glioma tumor growth. clinical validation aspects," *Applied Mathematics and Computation*, vol. 218, pp. 8779–8799, 05 2012.

[11] M. S. Alber, M. A. Kiskowski, J. A. Glazier, and Y. Jiang, "On cellular automaton approaches to modeling biological cells," pp. 1–39, 2003.

[12] G. Ermentrout and L. Edelstein-Keshet, "Cellular automata approaches to biological modeling," *Journal of Theoretical Biology*, vol. 160, no. 1, pp. 97 – 133, 1993.

[13] M. Amos, *Cellular computing.* Oxford University Press, 2020.

[14] Z. Wang, J. D. Butner, R. Kerketta, V. Cristini, and T. S. Deisboeck, "Simulating cancer growth with multiscale agent-based modeling," *Seminars in Cancer Biology*, vol. 30, pp. 70 – 78, 2015. Cancer modeling and network biology.

[15] G. An, Q. Mi, J. Dutta-Moscato, and Y. Vodovotz, "Agent-based models in translational systems biology," *WIREs Systems Biology and Medicine*, vol. 1, no. 2, pp. 159–171, 2009.

[16] M. Soheilypour and M. R. K. Mofrad, "Agent-based modeling in molecular systems biology," *BioEssays*, vol. 40, no. 7, p. 1800020, 2018.

[17] M. Hwang, M. Garbey, S. A. Berceli, and R. Tran-Son-Tay, "Rule-based simulation of multi-cellular biological systems—a review of modeling techniques," *Cellular and Molecular Bioengineering*, vol. 2, pp. 285–294, Sep 2009.

[18] K. Marias, V. Sakkalis, A. Roniotis, C. Farmaki, G. Stamatakos, D. Dionysiou, S. Giatili, N. Uzunoglou, N. Graf, R. Bohle, E. Messe, P. V. Coveney, S. Manos, S. Wan, A. Folarin, S. Nagl, P. Büchler, T. Bardyn, M. Reyes, G. Clapworthy, N. Mcfarlane, E. Liu, T. Bily, M. Balek, M. Karasek, V. Bednar, J. Sabczynski, R. Opfer, S. Renisch, and I. C. Carlsen, "Clinically oriented translational cancer multilevel modeling: The contracancrum project," pp. 2124–2127, 2010.

[19] A. Roniotis, K. Marias, V. Sakkalis, I. Karatzanis, and M. Zervakis, "The mathematical path to develop a heterogeneous, anisotropic and 3-dimensional glioma model using finite differences," pp. 1–4, 2009.

[20] V. Sakkalis, A. Roniotis, C. Farmaki, I. Karatzanis, and K. Marias, "Evaluation framework for the multilevel macroscopic models of solid tumor growth in the glioma case," pp. 6809–6812, 2010.

[21] A. Roniotis, K. Marias, V. Sakkalis, G. Stamatakos, and M. Zervakis, "Comparing finite elements and finite differences for developing diffusive models of glioma growth," pp. 6797–6800, 2010.

[22] T. S. Deisboeck and G. Stamatakos, *In Silico Oncology Part II - Clinical Requirements Regarding In Silico Oncology.* CRC Press, 2011.

[23] G. S. Stamatakos and D. D. Dionysiou, "Introduction of hypermatrix and operator notation into a discrete mathematics simulation model of malignant tumour response to therapeutic schemes in vivo. some operator properties," *Cancer Informatics*, vol. 7, p. CIN.S2712, 2009. PMID: 20011462.

[24] G. S. Stamatakos, E. Kolokotroni, D. Dionysiou, E. Georgiadi, and S. Giatili, "In silico oncology: a top-down multiscale simulator of cancer dynamics. studying the effect of symmetric stem cell division on the cellular constitution of a tumour," pp. 1830–1833, 2010.

[25] Retrieved 20 January 2021 from https://commons.wikimedia.org/wiki/File:Cell_Cycle_2-2.svg.

[26] G. M. Cooper, *The cell : a molecular approach.* ASM Press, 2 ed., 2000.

[27] J. A. Smith and L. Martin, "Do cells cycle?," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 70, pp. 1263–1267, Apr 1973. 4515625[pmid].

[28] R. S. Wu and W. M. Bonner, "Separation of basal histone synthesis from s-phase histone synthesis in dividing cells," *Cell*, vol. 27, pp. 321–330, Dec 1981.

[29] D. M. Nelson, X. Ye, C. Hall, H. Santos, T. Ma, G. D. Kao, T. J. Yen, J. W. Harper, and P. D. Adams, "Coupling of dna synthesis and histone synthesis in s phase independent of cyclin/cdk2 activity," *Molecular and cellular biology*, vol. 22, pp. 7459–7472, Nov 2002.

[30] I. L. CAMERON and R. C. GREULICH, "Evidence for an essentially constant duration of dna synthesis in renewing epithelia of the adult mouse," *The Journal of cell biology*, vol. 18, pp. 31–40, Jul 1963.

[31] A. Maton, *Cells: Building Blocks of Life.* Prentice-Hall, 1997.

[32] Retrieved 20 January 2021 from https://commons.wikimedia.org/wiki/File:Mitosis_Stages.svg.

[33] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher, "Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization," *Molecular biology of the cell*, vol. 9, pp. 3273–3297, Dec 1998.

[34] T. Pramila, W. Wu, S. Miles, W. S. Noble, and L. L. Breeden, "The forkhead transcription factor hcm1 regulates chromosome segregation genes and fills the s-phase gap in the transcriptional circuitry of the cell cycle," *Genes & development*, vol. 20, pp. 2266–2278, Aug 2006.

[35] D. A. Orlando, C. Y. Lin, A. Bernard, J. Y. Wang, J. E. S. Socolar, E. S. Iversen, A. J. Hartemink, and S. B. Haase, "Global control of cell-cycle transcription by coupled cdk and network oscillators," *Nature*, vol. 453, pp. 944–947, Jun 2008.

[36] E. A. Nigg, "Cyclin-dependent protein kinases: Key regulators of the eukaryotic cell cycle," *BioEssays*, vol. 17, no. 6, pp. 471–480, 1995.

[37] V. Kumar, N. Fausto, A. K. Abbas, R. S. Cotran, and S. L. Robbins, *Pathological Basis of Disease.* Elsevier Saunders, 2004.

[38] B. O'Leary, R. S. Finn, and N. C. Turner, "Treating cancer with selective cdk4/6 inhibitors," *Nature Reviews Clinical Oncology*, vol. 13, pp. 417–430, Jul 2016.

[39] B. Bilgin, M. A. Sendur, D. Şener Dede, M. B. Akıncı, and B. Yalçın, "A current and comprehensive review of cyclin-dependent kinase inhibitors for the treatment of metastatic breast cancer," *Current Medical Research and Opinion*, vol. 33, no. 9, pp. 1559–1569, 2017. PMID: 28657360.

[40] Press release. NobelPrize.org. Nobel Prize Outreach AB 2021. Thu. 16 Sep 2021. https://www.nobelprize.org/prizes/medicine/2001/press-release.

[41] G. Stamatakos, "Top-down multilevel simulation of tumor response to treatment in the context of in silico oncology," 09 2010.

[42] G. Stamatakos, E. Kolokotroni, D. Dionysiou, E. Georgiadi, and C. Desmedt, "An advanced discrete state–discrete event multiscale simulation model of the response of a solid tumor to chemotherapy: Mimicking a clinical study," *Journal of Theoretical Biology*, vol. 266, no. 1, pp. 124 – 139, 2010. Retrieved 20 January 2021 from https://www.sciencedirect.com/science/article/abs/pii/S0022519310002547.

[43] G. Stamatakos, E. Georgiadi, N. Graf, E. Kolokotroni, and D. Dionysiou, "Exploiting clinical trial data drastically narrows the window of possible solutions to the problem of clinical adaptation of a multiscale cancer model," *PloS one*, vol. 6, p. e17594, 03 2011. Retrieved 20 January 2021 from https://www.researchgate.net/publication/50399814_Exploiting_Clinical_Trial_Data_Drastically_Narrows_the_Window_of_Possible_Solutions_to_the_Problem_of_Clinical_Adaptation_of_a_Multiscale_Cancer_Model.

[44] G. M. Vujanić, J. R. Apps, V. Moroz, F. Ceroni, R. D. Williams, N. J. Sebire, and K. Pritchard-Jones, "Nephrogenic rests in Wilms tumors treated with preoperative chemotherapy: The UK SIOP Wilms Tumor 2001 Trial experience," *Pediatric Blood & Cancer*, vol. 64, Nov. 2017.

[45] J. B. Beckwith, "New developments in the pathology of Wilms tumor," *Cancer Investigation*, vol. 15, no. 2, pp. 153–162, 1997.

[46] J. B. Beckwith, "Precursor lesions of Wilms tumor: clinical and biological implications," *Medical and Pediatric Oncology*, vol. 21, no. 3, pp. 158–168, 1993.

[47] J. B. Beckwith, "Wilms' tumor and other renal tumors of childhood: A selective review from the national wilms' tumor study pathology center," *Human Pathology*, vol. 14, no. 6, pp. 481–492, 1983.

[48] S. D. Popov, N. J. Sebire, and G. M. Vujanic, "Wilms' Tumour – Histology and Differential Diagnosis," in *Wilms Tumor* (M. M. van den Heuvel-Eibrink, ed.), Brisbane (AU): Codon Publications, 2016.

[49] A. Hossain and G. F. Saunders, "The human sex-determining gene <em>sry</em> is a direct target of <em>wt1</em> *," *Journal of Biological Chemistry*, vol. 276, pp. 16817–16823, May 2001.

[50] K.-D. Wagner, N. Wagner, G. Schley, H. Theres, and H. Scholz, "The wilms' tumor suppressor wt1 encodes a transcriptional activator of the class iv pou-domain factor pou4f2 (brn-3b)," *Gene*, vol. 305, no. 2, pp. 217–223, 2003.

[51] E. A. Rose, T. Glaser, C. Jones, C. L. Smith, W. H. Lewis, K. M. Call, M. Minden, E. Champagne, L. Bonetta, H. Yeger, and D. E. Housman, "Complete physical map of the wagr region of 11p13 localizes a candidate wilms' tumor gene," *Cell*, vol. 60, pp. 495–508, Feb 1990.

[52] M. J. COPPES, V. HUFF, and J. PELLETIER, "Denys-drash syndrome : relating a clinical disorder to genetic alterations in the tumor suppressor gene wt1," *The Journal of pediatrics*, 1993.

[53] C. Schwienbacher, A. Angioni, R. Scelfo, A. Veronese, G. A. Calin, G. Massazza, I. Hatada, G. Barbanti-Brodano, and M. Negrini, "Abnormal rna expression of 11p15 imprinted genes and kidney developmental genes in wilms tumor," *Cancer Research*, vol. 60, no. 6, pp. 1521–1525, 2000.

[54] V. Huff, "Wilms tumor genetics: A new, unx-pected twist to the story," *Cancer Cell*, vol. 11, no. 2, pp. 105–107, 2007.

[55] Retrieved 30 August 2021 from `https://omim.org/phenotypicSeries/PS194070?sort=inheritance&order=desc`.

[56] G. Sonn and L. M. Shortliffe, "Management of wilms tumor: current standard of care," *Nature Clinical Practice Urology*, vol. 5, pp. 551–560, Oct 2008.

[57] A. M. Davidoff, "Wilms' tumor," *Current opinion in pediatrics*, vol. 21, pp. 357–364, Jun 2009. 19417665[pmid].

[58] J. M. Kalish, L. Doros, L. J. Helman, R. C. Hennekam, R. P. Kuiper, S. M. Maas, E. R. Maher, K. E. Nichols, S. E. Plon, C. C. Porter, S. Rednam, K. A. P. Schultz, L. J. States, G. E. Tomlinson, K. Zelley, and T. E. Druley, "Surveillance recommendations for children with overgrowth syndromes and predisposition to wilms tumors and hepatoblastoma," *Clinical cancer research : an official journal of the American Association for Cancer Research*, vol. 23, pp. e115–e122, Jul 2017.

[59] A. N. Ali, R. Diaz, H.-K. Shu, A. C. Paulino, and N. Esiashvili, "A surveillance, epidemiology and end results (seer) program comparison of adult and pediatric wilms' tumor," *Cancer*, vol. 118, no. 9, pp. 2541–2551, 2012.

[60] E. Revazova and A. Petrova, "[cell cycle and proliferative pool of human tumor strains transplanted into athymic mice]," *Biulleten' eksperimental'noi biologii i meditsiny*, vol. 92, p. 335—337, September 1981.

[61] K. Måseide and E. K. Rofstad, "Mathematical modeling of chronical hypoxia in tumors considering potential doubling time and hypoxic cell lifetime," *Radiotherapy and Oncology*, vol. 54, no. 2, pp. 171–177, 2000.

[62] W. Düchting, W. Ulmer, R. Lehrig, T. Ginsberg, and E. Dedeleit, "Computer simulation and modelling of tumor spheroid growth and their relevance for optimization of fractionated radiotherapy," *Strahlentherapie und Onkologie : Organ der Deutschen Rontgengesellschaft ... [et al]*, vol. 168, p. 354—360, June 1992.

[63] A. R. Anderson, A. M. Weaver, P. T. Cummings, and V. Quaranta, "Tumor morphology and phenotypic evolution driven by selective pressure from the microenvironment," *Cell*, vol. 127, no. 5, pp. 905–915, 2006.

[64] L. M. Wein, J. E. Cohen, and J. T. Wu, "Dynamic optimization of a linear–quadratic model with incomplete repair and volume-dependent sensitivity and repopulation," *International Journal of Radiation Oncology*Biology*Physics*, vol. 47, no. 4, pp. 1073–1083, 2000.

[65] B. Ribba, T. Colin, and S. Schnell, “A multiscale mathematical model of cancer, and its use in analyzing irradiation therapies,” *Theoretical Biology and Medical Modelling*, vol. 3, p. 7, Feb 2006.

[66] W. C. Dewey, C. C. Ling, and R. E. Meyn, “Radiation-induced apoptosis: Relevance to radiotherapy,” *International Journal of Radiation Oncology*Biology*Physics*, vol. 33, no. 4, pp. 781–796, 1995.

[67] E. Groninger, T. Meeuwsen-de Boer, P. Koopmans, D. Uges, W. Sluiter, A. Veerman, W. Kamps, and S. de Graaf, “Pharmacokinetics of vincristine monotherapy in childhood acute lymphoblastic leukemia,” *Pediatric Research*, vol. 52, pp. 113–118, Jul 2002.

[68] W. N. Dahl, R. Oftebro, E. O. Pettersen, and T. Brustad, “Inhibitory and cytotoxic effects of oncovin (vincristine sulfate) on cells of human line nhik 3025,” *Cancer Research*, vol. 36, no. 9 Part 1, pp. 3101–3105, 1976.

[69] K. Sawada, K. Noda, H. Nakajima, N. Shimbara, Y. Furuichi, and M. Sugimoto, “Differential cytotoxicity of anticancer agents in pre- and post-immortal lymphoblastoid cell lines,” *Biological and Pharmaceutical Bulletin*, vol. 28, no. 7, pp. 1202–1207, 2005.

[70] G. J. Veal, M. Cole, J. Errington, A. Parry, J. Hale, A. D. Pearson, K. Howe, J. C. Chisholm, C. Beane, B. Brennan, F. Waters, A. Glaser, S. Hemsworth, H. McDowell, Y. Wright, K. Pritchard-Jones, R. Pinkerton, G. Jenner, J. Nicholson, A. M. Elsworth, A. V. Boddy, and U. P. W. Group, “Pharmacokinetics of dactinomycin in a pediatric patient population: a united kingdom children’s cancer study group study,” *Clinical Cancer Research*, vol. 11, no. 16, pp. 5893–5899, 2005.

[71] E. C. Georgiadi, D. D. Dionysiou, N. Graf, and G. S. Stamatakos, “Towards in silico oncology: Adapting a four dimensional nephroblastoma treatment model to a clinical trial case based on multi-method sensitivity analysis,” *Computers in Biology and Medicine*, vol. 42, no. 11, pp. 1064–1078, 2012.

[72] https://ark.intel.com/content/www/us/en/ark/products/86067/intel-xeon-processor-e5-2658a-v3-30m-cache-2-20-ghz.html.

[73] https://www.valgrind.org/.

[74] https://valgrind.org/docs/manual/ms-manual.html.

[75] https://valgrind.org/docs/manual/cl-manual.html.

[76] https://valgrind.org/docs/manual/cg-manual.html.

[77] https://www.openmp.org/.

[78] Retrieved 20 September 2021 from `https://commons.wikimedia.org/wiki/File:AmdahlsLaw.svg`.

[79] Retrieved 2 September 2021 from `https://commons.wikimedia.org/w/index.php?curid=27378784`.

[80] https://kcachegrind.github.io/html/Home.html.