

# On the road to digital twins of tumors

Inaugural-Dissertation

zur Erlangung des Doktorgrades  
der Mathematisch-Naturwissenschaftlichen Fakultät  
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

**Eric Behle**  
aus Herrenberg

Düsseldorf, Januar 2025

aus dem Jülich Supercomputing Center (JSC)  
des Forschungszentrum Jülich

Gedruckt mit der Genehmigung der  
der Mathematisch-Naturwissenschaftlichen Fakultät  
der Heinrich-Heine-Universität Düsseldorf

Berichterstattende

1. Prof. Dr. Gunnar Schröder
2. Dr. Alexander Schug

Tag der mündlichen Prüfung: 21.05.2025

## Erklärung

Ich versichere an Eides statt, dass die Dissertation von mir selbständig und ohne unzulässige fremde Hilfe unter Beachtung der “Grundsätze zur Sicherung guter wissenschaftlicher Praxis an der Heinrich-Heine-Universität Düsseldorf” erstellt worden ist. Die Dissertation wurde in dieser oder ähnlicher Form noch bei keiner anderen Institution eingereicht. Ferner habe ich bislang keine erfolglosen oder erfolgreichen Promotionsversuche unternommen.

\_\_\_\_\_ Datum

\_\_\_\_\_ Eric Behle

# Abstract

Many multicellular organisms are composed of tissues. Tissues are groups of similar cells with one or more specific tasks, and it is the combination of multiple tissues that leads to the formation of organs and larger structures within the body. As the complexity of an organism increases, so does the number of different tissues. Tissue generation and differentiation thus plays crucial roles covering the range from life processes from its early stages during embryogenesis to necessary maintenance such as wound healing. Unfortunately, the proliferation machinery of some cells can become defective due to multiple causes. Through this, the tissue growth process may become flawed and uncontrollable. This can lead to the multi-faceted phenomenon known as cancer. Cancer is a disease that can occur in the entire body, and its heterogeneity makes a generally applicable treatment difficult. While understanding the detailed origins of the cell proliferation defects on a genetic and epigenetic level is crucial to stopping the disease before it starts, doctors and patients are often faced with the fact that it has already progressed to a macroscopic stage ( $O(\text{mm})$ ) at the time of diagnosis. In this situation, it is necessary to understand the laws governing the growth of tumors, in order to develop effective treatment strategies. Computational models are a useful tool for this for two main reasons. Firstly, because a mechanistic model capable of reproducing experimentally observed behavior underlines our understanding of the underlying mechanisms. Secondly, because a working model would vastly benefit treatment via quick assessments of options without any side effects or harm to the patient. Hence, work is ongoing to develop so-called digital twins of growing tumors. A challenge in this endeavour is the scale-spanning nature of cancer. This means that any model simulating macroscopic tissue and tumor growth must be capable of doing so at single-cell resolution. Such models are computationally demanding, and often require supercomputing infrastructures to employ. Furthermore, care must be taken to parameterize them correctly, and large amounts of experimental data are required for this. During this doctoral project, I have worked with and extended *Cells in Silico* (CiS), a highly scalable tissue simulation framework previously developed by my group. CiS is capable of simulating biological tissues composed of tens of millions of cells at subcellular resolution, and is therefore

a promising candidate for simulating a digital twin. However, before doing so, it must be extended further, and data for its parameterization must be found. To combat the problem of *in vivo* data scarcity, I have employed a "divide and conquer" approach, in which I aimed to partially parameterize CiS by focusing on smaller *in vitro* sub systems, for which data exist. During my studies, I first focused on an investigation of the structural environment of tumors, by working with tumor spheroids grown in collagen matrices of varying density. For this, I performed a large number of spheroid growth simulations, in order to reproduce the behavior of the *in vitro* spheroids. To analyze the agreement between *in vitro* and *in silico* spheroids, we developed the overall deviation score (ODS). The ODS, which is a metric for comparing the structure of two spheroids regardless of their origin, provided an objective function for the parameterization of CiS. During this project, we discovered that CiS needs a more realistic description for the extracellular matrix in order to accurately reproduce spheroid behavior. A project to include such a description is ongoing within my group. In the second part of my project, I focused on the nutrient environment of tumors. Here, I incorporated a set of detailed mouse brain vasculature data into CiS, in order to build a more realistic nutrient environment. I then studied the growth behavior of tumors placed in vascular environments of different density and thickness. Within my simulations, I found that vessel density is the main contributor to final tumor volume. Finally, I focused on the advancement of supercomputing infrastructure by participating in the development of a benchmarking pipeline for the JUPITER supercomputer. Overall, my work has improved CiS, and paved the way for bringing it closer to simulating digital twins of tumors.

# Zusammenfassung

Viele multizelluläre Organismen bestehen aus verschiedenen Geweben. Gewebe bestehen aus Gruppierungen ähnlicher Zellen mit einer oder mehreren spezifischen Aufgaben, und die Kombination mehrerer Gewebetypen führt zu der Formation von Organen und größeren Strukturen im Körper. Während die Komplexität eines Organismus sich erhöht, erhöht sich auch die Menge unterschiedlicher Gewebearten, und Gewebewachstum und -differenzierung sind unabdinglich sowohl für Embryogenese als auch Wundheilung. Leider kann die Teilungsmaschinerie von manchen Zellen aus vielerlei Gründen fehlerhaft werden. Hierdurch kann das Gewebewachstum fehlerhaft und unkontrolliert verlaufen, was zu dem facettenreichen Phänomen Krebs führt. Krebs ist eine Krankheit, die im gesamten Körper auftreten kann, und seine Heterogenität macht eine global anwendbare Behandlungsstrategie schwierig. Während es für die Prävention von Krebskrankheiten entscheidend ist, die detaillierten Ursprünge der Fehler in der Teilungsmaschinerie auf genetischer und epigenetischer Ebene zu verstehen, sind Ärzte und Patienten leider häufig mit der Tatsache konfrontiert, dass die Krankheit zur Zeit der Diagnose bereits vorangeschritten ist. In dieser Situation ist es nötig, die Gesetze des Wachstums von Tumoren zu verstehen, um effektive Behandlungsstrategien zu entwickeln. Computergestützte Modelle sind hier aus zwei Hauptgründen ein nützliches Werkzeug. Erstens, weil ein mechanistisches Modell, das in der Lage ist, experimentell beobachtetes Verhalten zu reproduzieren, unser Verständnis der unterliegenden Mechanismen unterstreicht. Zweitens, weil ein funktionierendes Modell die schnelle Evaluation von Behandlungen erlaubt, ohne dem Patienten zu schaden. Daher ist die Entwicklung sogenannter digitaler Zwillinge von Tumoren ein Bereich aktiver Forschung. Eine Komplikation in diesem Bemühen ist die skalenübergreifende Natur des Phänomens Krebs. Diese bedeutet, dass ein Modell, welches makroskopisches Gewebe und Tumorwachstum simuliert, in der Lage sein muss, dies auf zellulärer Auflösung zu tun. Solche Modelle sind sehr rechenintensiv und müssen daher oft auf Superrechnern laufen. Desweiteren muss die Parametrisierung eines solchen Modells korrekt durchgeführt werden, und hierfür sind große Mengen experimenteller Daten erforderlich. Während dieses Promotionsprojekts habe ich das Gewebesimulationsmodell *Cells in Silico* (CiS)

verwendet und weiterentwickelt. CiS ist in der Lage, biologisches Gewebe bestehend aus vielen Millionen von Zellen bei subzellulärer Auflösung zu simulieren, und ist daher ein vielversprechender Kandidat für die Simulation digitaler Zwillinge. Um dies zu tun, muss es allerdings weiter entwickelt werden, und Daten für seine Parametrisierung müssen gefunden werden. Um das Problem des Mangels an *in vivo* Daten zu umgehen, habe Ich einen Ansatz verwendet, in dem ich mich auf kleinere *in vitro* subsysteme fokussiert habe, für die Daten existieren. Während meiner Studien habe ich mich zunächst mit der strukturellen Umgebung von Tumoren beschäftigt, indem ich mit Tumor Spheroide gearbeitet habe, die in kollagenhaltigen Medien verschiedener Dichte gezüchtet wurden. Hierfür habe ich eine große Menge an Simulationen von Spheroidwachstum erstellt, um das Verhalten der *in vitro* Spheroiden zu reproduzieren. Um die Übereinstimmung zwischen *in vitro* und *in silico* Spheroide zu analysieren, haben wir den overall deviation score (ODS) entwickelt. Der ODS, der eine Metrik für den Vergleich der Struktur zweier Spheroiden unabhängig ihrer Herkunft darstellt, konnte als Zielfunktion für die Parametrisierung von CiS verwendet werden. Während dieses Projekts fanden wir heraus, dass CiS eine realistischere Beschreibung der extrazellulären Matrix benötigt, um das Verhalten von Spheroide akkurat zu reproduzieren. Ein Projekt mit dem Ziel, dies zu erreichen, läuft zurzeit in meiner Arbeitsgruppe. Im zweiten Teil meines Projekts habe ich mich auf die Nährstoff-Umgebung von Tumoren fokussiert. Hier habe ich ein Set von hochaufgelösten Daten der Blutgefäßstruktur eines Mäusegehirns in CiS eingebunden, um ein realistischeres Nährstoffumfeld zu generieren. Dann habe ich das Wachstumsverhalten von simulierten Tumoren untersucht, die in Blutgefäßsysteme verschiedener Dichte und Dicke platziert wurden. Innerhalb meiner Simulationen stellte sich heraus, dass die Blutgefäßdichte den Hauptbeitrag zur finalen Tumorgröße geleistet hat. Zuletzt habe ich zu Weiterentwicklung von Hochleistungsrechnensystemen beigetragen, indem ich an einer Benchmark Pipeline für den JUPITER Supercomputer teilgenommen habe. Zusammenfassend hat meine Arbeit CiS weiterentwickelt, und den Weg zur zukünftigen Simulation digitaler Zwillinge von Tumoren geebnet.

# Acknowledgement

I want to thank everyone who helped and supported me in completing this dissertation. First of all, I want to thank Alexander Schug for supervising and mentoring me. I also want to thank Gunnar Schröder for agreeing to be my first supervisor at HHU. Furthermore, I want to thank Adélaïde Raguin for agreeing to act as co-supervisor, and for the deep scientific discussions we had.

I spent a lot of time working on projects together with my colleague Julian Herold. I want to thank him for the fruitful collaborative work and long hours working on code and papers together.

My thanks also go to my colleague Christian Faber, who started his doctoral studies at the same time as me, and who has been a great discussion partner and desk neighbor.

I want to thank Jakob Rosenbauer and Marco Berghoff for getting me started with the project and answering my many initial questions.

To my colleagues Utkarsh Upadhyay, Anton Dorn, Ali Sharifian, Oskar Taubert and Fathia Idiris, I want to express my thanks for the great work group environment over the years.

Finally, I want to thank my partner Pauline, who has supported me and been at my side throughout this time.

# List of publications

This thesis includes peer-reviewed articles and submitted manuscripts about the work undertaken during my doctoral project.

## Published:

- Julian Herold\*, **Eric Behle\***, Jakob Rosenbauer, Jacopo Ferruzzi, Alexander Schug, *Development of a scoring function for comparing simulated and experimental tumor spheroids*, PLOS Computational Biology, 2023

*\*Shared first authors*

- Andreas Herten, Sebastian Achilles, Damian Alvarez, Jayesh Badwaik, **Eric Behle**, Mathis Bode, Thomas Breuer, Daniel Caviedes-Voullième, Mehdi Cherti, Adel Dabah, Salem El Sayed, Wolfgang Frings, Ana Gonzalez-Nicolas, Eric B. Gregory, Kaveh Haghighi Mood, Thorsten Hater, Jenia Jitsev, Chelsea Maria John, Jan H. Meinke, Catrin I. Meyer, Pavel Mezentsev, Jan-Oliver Mirus, Stepan Nassyr, Carolin Penke, Manoel Römmer, Ujjwal Sinha, Benedikt von St. Vieth, Olaf Stein, Estela Suarez, Dennis Willsch, Ilya Zhukov, *Application-Driven Exascale: The JUPITER Benchmark Suite*, SC24, 2024

## Preprint:

- **Eric Behle**, Julian Herold, Alexander Schug, *Leveraging Experimental Vasculature Data for High Resolution Brain Tumor Simulations*, bioRxiv, 2024

# Contents

<b>1</b>	<b>Introduction and background</b>	<b>12</b>
1.1	Cancer . . . . .	13
1.1.1	Hallmarks . . . . .	14
1.1.2	Enabling characteristics . . . . .	15
1.1.3	Cancer treatment . . . . .	16
1.1.4	<i>In vitro</i> cancer model systems . . . . .	16
1.2	Simulation of cancerous tissue . . . . .	17
1.2.1	Continuum models . . . . .	17
1.2.2	Agent based models . . . . .	18
1.2.3	Cells in Silico . . . . .	20
1.3	Divide and conquer . . . . .	23
<b>2</b>	<b>Results</b>	<b>25</b>
2.1	Tumor structural environment . . . . .	25
2.2	Tumor nutrient environment . . . . .	27
2.2.1	Static vasculature: mouse brain data . . . . .	27
2.2.2	Dynamic vasculature: angiogenesis . . . . .	27
2.3	Numerical aspects . . . . .	34
2.3.1	Pitfalls in HPC development . . . . .	34
2.3.2	Boundary exchange considerations . . . . .	34
2.3.3	Parallel IO . . . . .	35
2.3.4	JUPITER benchmark . . . . .	36
<b>3</b>	<b>Conclusion and outlook</b>	<b>38</b>
	<b>Appendices</b>	<b>55</b>
<b>A</b>	<b>Embedded Publication 1</b>	<b>56</b>
<b>B</b>	<b>Embedded Publication 2</b>	<b>83</b>
<b>C</b>	<b>Embedded Publication 3</b>	<b>103</b>

# List of abbreviations

**TME** Tumor microenvironment

**PDE** Partial differential equation

**HPC** High performance computing

**CPM** Cellular Potts model

**CiS** *Cells in Silico*

**NAStJA** *Neoteric Autonomous Stencil code for Jolly Algorithms*

**ECM** Extracellular matrix

**VEGF** Vascular Endothelial Growth Factor

**ATP** Adenosine triphosphate

**ABM** Agent Based model

**BVS** Blood Vessel Segment

# Chapter 1

## Introduction and background

Cancer is one of the main remaining diseases plaguing humanity [1]. Contrary to many other illnesses for which we now have effective cures, this one remains elusive [2, 3]. This is due to the fact that cancer is not a singular ailment with a specific weakness, but rather denotes a collection of afflictions arising from multiple causes [4, 5, 6]. Each of these causes brings its own complexity, and the interactions between them add further to the difficulty in finding generally applicable treatments. Thus, studying and treating cancer presents a significant challenge. Undeterred by this, scientists continue to probe the many aspects of the disease. To do so, the problem is tackled from multiple angles. An important pillar of this is the extensive experimental work that continues to be done in this field. Here, researchers utilize a wide range of methods. Single cells are studied using novel high-throughput sequencing [7] or motility tracking [8]. On a larger scale, *in vitro* tissue arrangements such as tumor spheroids [9] and tissue organoids [10] are used to study the behavior of cancerous cells in various conditions. On the tissue scale, the advance of *in vivo* treatments is also continuing [11]. The importance of the application and advancement of such experimental methods remains. However, a crucial way to ensure our understanding of the effects which we observe experimentally is the computational study of the available data and the development of mechanistic models. Much progress has been made, for example, on the side of driver gene detection via whole genome analysis [12]. The application of ML-based data analysis methods is also increasing [13]. A multitude of mechanistic tumor models exist, which have been successfully applied to simulate aspects of tumor growth [14, 15, 16, 17, 18, 19]. Hence, these additional pillars represent a set of complementary approaches of growing importance. In particular, the prospect of patient-tailored digital twins of tumors is an exciting branch of research [18]. Digital twins, which denote exact *in silico* representations of *in vitro* or *in vivo* tumors, are promising, because they allow for the extensive testing of treatments without the danger of causing harm to the patient through ineffectual options.

Furthermore, results of treatments can be obtained much faster due to the fact that months of tumor growth can be simulated within hours or days. This further improves their ability of high-throughput testing. The main drawback of such computational models is the fact that their development requires extensive hardware resources [17]. Specifically, the dynamics both at the single-cell level and the tissue level need to be incorporated to accurately model tumor behavior [20]. Hence, scale bridging models are required, and such models necessarily bring challenges on the side of computational efficiency [17]. A further challenge to overcome is the issue of model parameterization, which requires extensive amounts of data. The ideal datasets for the parameterization of digital twins would include, for each cell within the tissue, time-resolved details on cell behavior, such as metabolic activity, signal transduction, and location and movement within the tissue. Metabolic activity such as glucose metabolism can be measured on a coarse level using PET imaging [21], but such methods do not resolve single cells. In addition, cell tracking within live tissue is a monumental challenge, the methods for which are still under development [22]. While the amount of data is increasing [23], and they are expected to be applicable in the medium term [24], other strategies must be considered during the current window of data scarcity. During the course of this doctoral thesis, I explored the two challenges of applying a scale-bridging model and data scarcity.

This document is structured as follows. First, I will present a summary of the main aspects of cancer, including current treatments and *in vitro* model systems. Next, I will highlight some common strategies for tissue simulation, and will then describe our simulation framework *Cells in Silico* (CiS) in detail. Following this, I will present my results in three sub topics: a study of the structural environment of tumors, a study of the nutrient environment of tumors, and numerical aspects related to working within a high-performance computing (HPC) environment. Finally, I will discuss the insights gained, reflect on the progress on the path to a full digital twin, and present some possible studies for the future.

## 1.1 Cancer

Brown et al. use the following definition to define cancer: "Cancer is a disease of uncontrolled proliferation by transformed cells subject to evolution by natural selection" [25]. Translating this to a larger scale, the term cancer groups a multitude of tissue aberrations, arising from wrongly behaving cells. This can occur in virtually any part of the body [25]. The ten most commonly found cancers are lung, breast, prostate, colon, stomach, liver, rectal, esophagus, cervix uteri and thyroid cancer [26]. Of these, the ones with the highest mortality rate are

lung, liver and stomach cancer [26]. Hanahan and Weinberg have identified eight "hallmarks" and three "enabling characteristics", which are common to all cancers [4, 5, 6]. They are briefly highlighted below.

### 1.1.1 Hallmarks

A hallmark refers to a phenomenological description of a behavior exhibited by a cancerous cell or the cancerous tissue it is part of. Hallmarks can have different origins depending on the type of cancer.

**Self sufficiency in growth signals** Within multicellular organisms, cell division is usually governed by external stimuli [27]. As a consequence, healthy cells only divide when they receive signals to do so from their surrounding [28]. One important aspect of cancerous cells is the fact that they are capable of mimicking these external stimuli. They achieve this, for example, by producing mitotic signaling molecules that affect themselves as well as other cells [29]. Alternatively, the receptors on the cell surface are increased such that the cell is more responsive to external stimuli [30].

**Insensitivity to anti-growth signals** In addition to being self sufficient in growth, cancerous cells also need to acquire an immunity to anti-growth signals which may be released by the body to fight them [31]. The way how this immunity is gained is either through the removal of receptors for growth-inhibiting signal molecules, or through the disruption of intracellular pathways triggered by them. [32].

**Evasion of apoptosis** Apoptosis denotes the process of induced cell death. This either occurs as a result of external signaling, or is triggered by internal failsafe mechanisms [33]. Similar to the evasion of anti-growth signals, cancerous cells must also be able to evade apoptotic signals [34]. These two hallmarks are therefore closely related but distinct in their goal.

**Limitless replicative potential** Any cell is a product of one or more differentiation steps originating from a dividing stem cell of equal or higher order (e.g. multipotent, pluripotent, omnipotent) [35]. Upon reaching terminally differentiated state, a cell is usually not able to divide further, and eventually reaches senescent state [36]. Even those cells which continue to divide eventually reach senescence, usually when their telomeres become too short [37]. In order to reach limitless replicative potential, a cancer cell must both re-activate its division machinery and ensure its telomeres remain sufficiently long [38, 39].

**Sustained angiogenesis** A growing tumor requires ever more nutrients to proliferate. Once it reaches a critical size, the nutrient influx provided by the existing vascular network surrounding the tumor becomes insufficient [40]. In this case, the tumor can only proliferate further by inducing growth of new blood vessels towards it. This process, known as tumor-induced angiogenesis, is achieved through the release of pro-angiogenic signals, such as VEGF, by hypoxic cells in the center of the tumor [40, 41]. Blood vessels then grow along the gradient of this signal [42].

**Tissue invasion and metastasis** In addition to inducing angiogenesis to alter the immediate nutrient environment, cancerous cells may eventually leave the tumor and invade the surrounding tissue [43]. Once such cells reach locations in which they can thrive, they form secondary tumors [44]. Tumor-induced angiogenesis is often accompanied by subsequent metastasis, because tumor cells then have easy access to the entire body by traveling through the blood stream [45].

**Avoiding immune destruction** One of the tasks of the immune system is to recognize and target irregularly behaving cells within the body, which are then destroyed [46]. Hence, the tumor microenvironment (TME) is usually in a chronically inflamed state that contains many immune cells [47]. Unfortunately, this also produces a selective pressure on cell mutants capable of evading detection by the immune cells [48]. These can then freely divide further until the tumor population is dominated by immune-evading cells. Evasion is achieved by targeting antigen recognition or by altering the metabolism of T cells [49].

**Deregulating cellular energetics** Cancer cells often rely on anaerobic glycolysis for the production of ATP, instead of the more efficient oxydative phosphorylation found in healthy tissue [50, 51]. There is evidence that this is beneficial for survival in the TME, since it is highly heterogeneous, and even vascularized tumors contain hypoxic areas [52]. Furthermore, glycolysis generates more byproducts used for biosynthesis than oxydative phosphorylation and may therefore facilitate faster cell proliferation [5].

### 1.1.2 Enabling characteristics

The hallmarks mentioned above are only a summary of behaviors attributed to cancerous tissue. In order to acquire these behaviors, either the cell's genome or gene expression profile, i.e. epigenetics, need to be changed. Several enabling characteristics govern this process.

**Genome instability and mutation** In order for hallmark-enabling genome mutations to occur within a human lifetime, the rate of mutation must be high enough [4]. Usually, the DNA repair mechanisms of cells ensure that this is not the case, and in order for mutations to occur fast enough, these mechanisms must first be sabotaged [53]. A prominent example of this is the p53 tumor suppressor protein, whose function is lost in most tumors [4].

**Tissue inflammation** As mentioned in the *Avoiding immune destruction* hallmark, it has been found that the TME is in a state of permanent inflammation [47]. Here, the actions of the immune system create selective pressure which eventually leads to the occurrence of immune-evading cancer cell types. However, it has also been found that the conditions within non-cancerous inflamed tissue favor the acquisition of other hallmarks [54, 5].

**Nonmutational epigenetic reprogramming** More recently, the role of epigenetics has come more into the scientific focus. Aberrantly methylated regions [55] can induce expression of different proteins, and hence the protein landscape of the cell is altered [56]. In this way, cancer hallmarks can arise even though the genome itself has not been changed. This is becoming more important as more mechanisms are being discovered [57].

### 1.1.3 Cancer treatment

Some cancers, such as Glioblastoma and pancreatic cancer are not yet effectively treatable [58, 59, 60]. For others, the three traditional treatment options are chemotherapy [61], radiotherapy [62] and surgery [63]. All three have advantages and disadvantages [64, 62, 65]. Therefore, a combination of multiple treatments is often used, especially since surgery may increase the chance of metastasis [65]. The optimal combination is an area of active research. However, treatment trials can only be performed in a limited fashion for a given patient, and here, *in vitro* and *in silico* studies may be promising [18].

### 1.1.4 *In vitro* cancer model systems

Cancer cells can be grown *in vitro*, and hence, experiments have been designed which investigate their behavior in various environments [66]. Here, I outline some commonly used model systems.

**Cancer cell lines** Cell lines have been extracted and cultured *in vitro* since the middle of the 20th century [67]. The HeLa cell line is the oldest one in current use

[68]. Cell lines can be grown in many different conditions, and therefore many scenarios can be designed to test their proliferative behavior and resilience. However, it is not clear, how well the results from such experiments can be translated into the *in vivo* context [69, 70]. This has two main reasons. Firstly, the older a cell line is, the more it deviates from the originally extracted cells[71]. Secondly, *in vitro* cells in traditional model systems exist in environments which likely do not reproduce the conditions found in *in vivo* tissues. Due to the strong influence of environmental cues on the behavior of cells, this may lead to different behavior[72].

**Tumor spheroids** To build an environment that is closer to real tissues, cells are often cultured in the form tumor spheroids [73, 74]. Tumor spheroids are spherical aggregates of a few hundred to a few million cells, which are placed into a structural matrix. This matrix is composed of structural proteins such as collagen, and closely resembles the ECM found in real tissue [9]. Tumor spheroids can be used to study the invasion behavior of cells depending on their surrounding [9]. They can also function as testing grounds for new chemotherapy or radiotherapy treatments [73, 75].

**Tissue organoids** While tumor spheroids usually originate from only a single cell type, tissue organoids represent an attempt to mimic real tissues more closely [76]. They are composed of multiple regions of different cell type, and therefore exhibit a heterogeneous environment [74]. Organoids as a system are still in development, but this is a strongly growing field [74, 10].

## 1.2 Simulation of cancerous tissue

Cancer is a disease that incorporates both single cell and tissue effects. Computational models therefore need to be capable of treating both scales in order to accurately describe tumor growth. In this section, I first outline commonly used models for simulating tissue. I briefly mention continuum models and mostly focus on agent based models that resolve individual cells. Finally I describe *Cells in Silico* (CiS), the model that I utilized and further developed during this dissertation.

### 1.2.1 Continuum models

One way of circumventing the computational cost of simulating many individual cells is approximating populations as a continuous volumes. Partial differential equations (PDEs) can then be used to describe nutrient flow, tissue expansion and other effects [77]. While it is possible to simulate heterogeneous tissues and tumors with this approach, and there has been some success [77, 78], behavior emerging

from single-cell effects, e.g. invasion by single cells, is not tracked. For this, agent based models are more promising.

### 1.2.2 Agent based models

In an agent-based model (ABM), a system is described by individual agents which act in an environment. Each agent has a set of properties that dictate how it behaves and how it interacts with its environment. In an ABM, multiple such agents exist, and while the strength of the properties of two agents can be similar, each has their own individual set. ABMs can be applied in many disciplines, and an agent can be, for example, a single cell [14], a person in a population model [79], or an enzyme digesting a polymer [80]. The advantage of ABMs is that emergent behavior can be studied by investigating the individual agent properties. On the other hand, they are computationally expensive at increasing numbers of agents. Furthermore, care must be taken to accurately describe the environment and the interactions between agents. In the context of tissue modeling for example, important agent attributes would be cell center, cell volume, adhesion strength to other cells, cell motility strength, division rate and mutation rate. Here, the interaction with other cells depends strongly on how the volume, surface and overall shape of cells are described. Hence, agent-based modeling is usually combined with other modeling frameworks, which describe the interactions between the agents [14, 81]. Some common ones are highlighted below (see also Figure 1.1).

**Cells as points** The simplest approach to model a tissue with cellular resolution is to treat every cell as a point. Cellular automata (CA), for example, are lattice models, in which each lattice point can be used to describe a single cell. The system then evolves via nearest neighbor interactions. A very simple CA model is Conway's game of life [82]. Unfortunately, we have no information on cellular surfaces or volumes using this approach.

**Cells as spheres** A slightly more complex formulation is the description of each cell agent as a sphere with a center  $\vec{r}$  and a radius  $R$ . The interaction between cells can be described utilizing a hard or, more common, a soft sphere potential. Some success has been achieved here here [83, 84]. However, this approach only rudimentarily includes cell deformation, which is an important feature distinguishing cells [85].

**Edge based models** In both the vertex model (VM) and the voronoi tessellation model (VTM), cells are described as polygons whose edges connect them to their neighbors (see Figure 1.1). These edges can change due to pressure changes or

other effects. The difference between the two models lies in how the edges are determined. In a VM we define the location of vertices, i.e. junction points of three or more cells, and the edges follow from this. In the VTM on the other hand we define the cell centers, and the edges are determined via Voronoi tessellation [86]. VMs have been used to simulate *Drosophila* wing disk formation [87]. They have also been used for studying tissue jamming transitions [88]. However, these models still do not enable realistic cell deformation.

**Cellular Potts models** A significantly more detailed approach to a lattice model is the Cellular Potts Model (CPM), developed by Graner and Glazier [89]. In the CPM, each cell is described as a connected region of lattice points which contain the same ID. It is an extension of the Potts model [90], which is itself an extension of the Ising Model [91]. While The Potts model generalizes the Ising model by introducing an arbitrary number of IDs instead of just 0 or 1, the CPM goes further by defining an additional "type" variable for each ID. Type-dependent adhesion interactions between regions are then implemented. In the CPM, we define the overall energy of the lattice with the following Hamiltonian [81]:

$$\begin{aligned}
H_{\text{CPM}} = & \\
& \sum_{c \in \mathcal{C}} \lambda_{V, \tau(c)} \left( v(c) - V(\tau(c)) \right)^2 && \text{Cell volumes} \\
& + \sum_{c \in \mathcal{C}} \lambda_{S, \tau(c)} \left( s(c) - S(\tau(c)) \right)^2 && \text{Cell surfaces} \\
& + \sum_{i \in \Omega} \sum_{j \in N(i)} J_{\tau(c_i), \tau(c_j)} \left( 1 - \delta_{c_i, c_j} \right). && \text{Cell-cell adhesion}
\end{aligned} \tag{1.1}$$

Here,  $\lambda_{V, \tau(c)}$  is the coupling factor of the volume energy contribution for the cell type  $\tau(c)$  of the cell  $c$ ,  $v(c)$  is the current volume of cell  $c$ ,  $V(\tau(c))$  is the target volume for cells of type  $\tau(c)$ ,  $\lambda_{S, \tau(c)}$  is the coupling factor of the surface energy contribution for the cell type  $\tau(c)$  of the cell  $c$ ,  $s(c)$  is the current surface of cell  $c$ ,  $S(\tau(c))$  is the target surface for cells of type  $\tau(c)$ ,  $i$  is the  $i$ -th point in the overall lattice  $\Omega$ ,  $j$  is the  $j$ -th neighbor of all neighbors  $N$  of point  $i$ ,  $J$  is the adhesion matrix defining the adhesion strengths between all cell types, and  $\delta_{c_i, c_j}$  is the Kronecker delta. This Hamiltonian can optionally be extended by further terms, e.g. for including cell motility [92]. The system is propagated via the Metropolis algorithm [93]. In each iteration, we perform randomized swaps of lattice IDs, and review their impact on the system energy. The energy difference

$$\Delta H_{\text{CPM}} = H_{\text{CPM, new}} - H_{\text{CPM, old}} \tag{1.2}$$

between the energy  $H_{\text{CPM,new}}$  after the swap and  $H_{\text{CPM,old}}$  before the swap is then used to determine acceptance or rejection of the swap. The probability of acceptance  $p_{\text{accept}}$  is

$$p_{\text{accept}} = \begin{cases} 1, & \text{if } \Delta H_{\text{CPM}} \leq 0. \\ e^{-\frac{\Delta H_{\text{CPM}}}{T}}, & \text{otherwise.} \end{cases} \quad (1.3)$$

where  $T$  is the system temperature. Thus, we obtain a probabilistic simulation, with which deformable cells can be described. One major advantage of the CPM is the fact that the individual energy contributions per lattice point depend only on nearest-neighbor interactions. This means that it is easy to parallelize such simulations by dividing the grid into multiple sub grids [81].

**Deformable cell model** Off-lattice models such as deformable cell models (DFMs) represent the cell surface at higher detail. They do this via surface triangulation similar to finite element methods [19]. While more accurate, they also come with high computational cost.

### 1.2.3 Cells in Silico

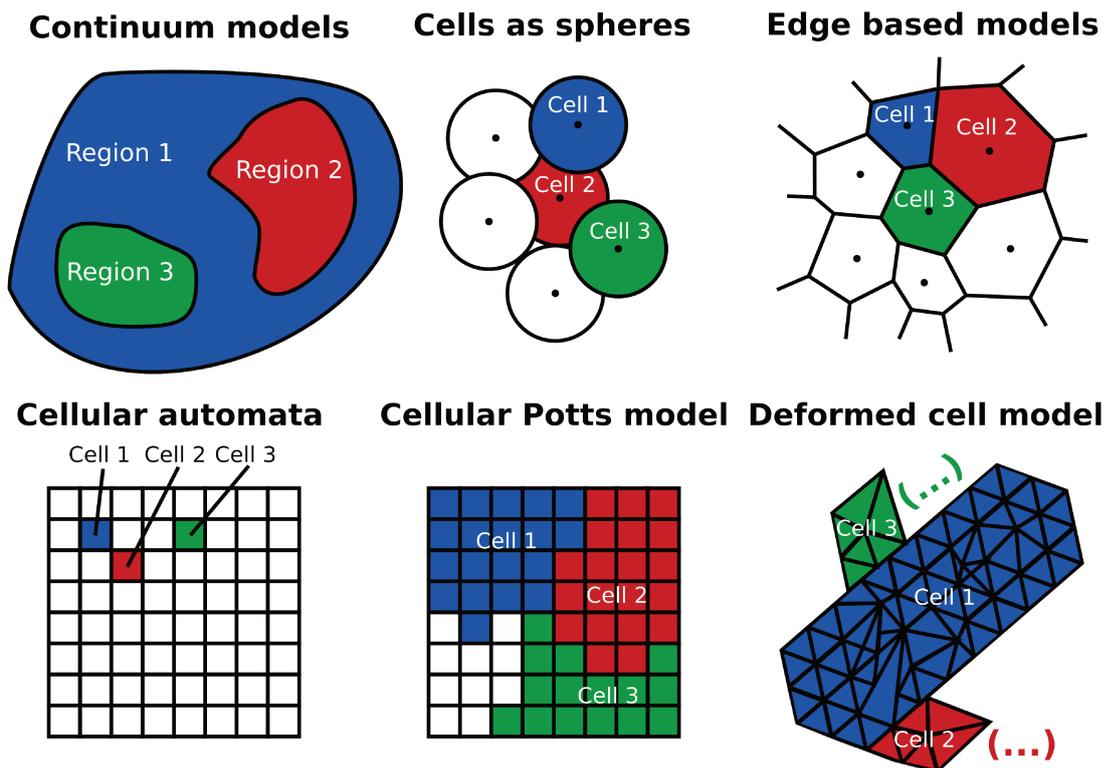
During this dissertation, I used and further developed a model called *Cells in Silico* (CiS) [81]. CiS is a highly parallelizable tissue simulation framework, which is an extension of the *NAStJA* stencil code solver [94, 95]. It is a hybrid model composed of three main layers, which I describe below.

**Microscale CPM layer** In CiS, cell shape and adhesion interactions are described by a parallelized CPM. The implemented CPM hamiltonian also includes a cell motility term [92, 96], so cells are capable of self-propelled movement within the system:

$$H_{\text{CPM, mot}} = H_{\text{CPM}} + \sum_{c \in C} \lambda_{\text{mot}, \tau(c)} \cdot \vec{m}_c \cdot \vec{R}_c \quad \text{Cell motility} \quad (1.4)$$

Here,  $\lambda_{\text{mot}, \tau(c)}$  is the coupling factor of the motility energy contribution for the cell type  $\tau(c)$  of the cell  $c$ ,  $\vec{m}_c$  is the current motility direction of cell  $c$ , and  $\vec{R}_c$  is its center of mass.  $\vec{m}_c$  can be periodically updated to introduce random or persistent random walk behavior.

Furthermore, the CPM lattice of CiS can contain so-called solids. These are lattice points, which are not included in the randomized swapping and are therefore not affected by the hamiltonian. They can be used to form structures such as a solid ECM or rigid blood vessels. Due to the excellent parallelizability of CiS, we



**Figure 1.1:** Sketch of various methods for the simulation of tissue. Continuum models describe tissues as continuous volumes, whose dynamics are described via PDEs. Spherical cell models assign a center and radius to each cell, and include hard- or soft sphere potentials between them. Edge based models track the cell centers and cell-cell boundaries. Cellular automata are simple lattice models in which each cell is a single lattice point. Cellular Potts models are more elaborate lattice models in which cells are described as connected lattice regions of same ID. In deformed cell models, each cell's surface is represented by triangulation similar to finite element methods.

can simulate tissues composed of tens of millions of cells at subcellular resolution using modern supercomputing systems.

**Mesoscale signal and nutrient interaction layer** A multitude of signaling molecules exist within biological tissues [97]. These signals govern a wide range of phenomena, from embryogenesis to immune response, cell differentiation, cell death etc. [97]. To include such effects, CiS therefore contains a simplified signal exchange functionality, in which signals can diffuse between adjacent cells. The diffusion flux  $F_{i,j}^k$  of the  $k$ -th signal between cell  $i$  and cell  $j$  is defined as follows:

$$F_{i,j}^k = D_{\tau(i),\tau(j)} \left( \frac{A_{i,j}}{A_i} + \frac{A_{i,j}}{A_j} \right) ([S]_j^k - [S]_i^k). \quad (1.5)$$

Here,  $D_{\tau(i),\tau(j)}$  is the type-dependent diffusion constant,  $A_{i,j}$  is the interface area between cell  $i$  and cell  $j$ ,  $A_i$  and  $A_j$  are the overall surface areas of cell  $i$  and cell  $j$ , and  $[S]_i^k$  and  $[S]_j^k$  are the signal concentrations within each cell [81]. Cells can both produce and consume signals, and solids in the CPM layer can function as sources or sinks with constant signal concentration, and thereby function as blood vessels.

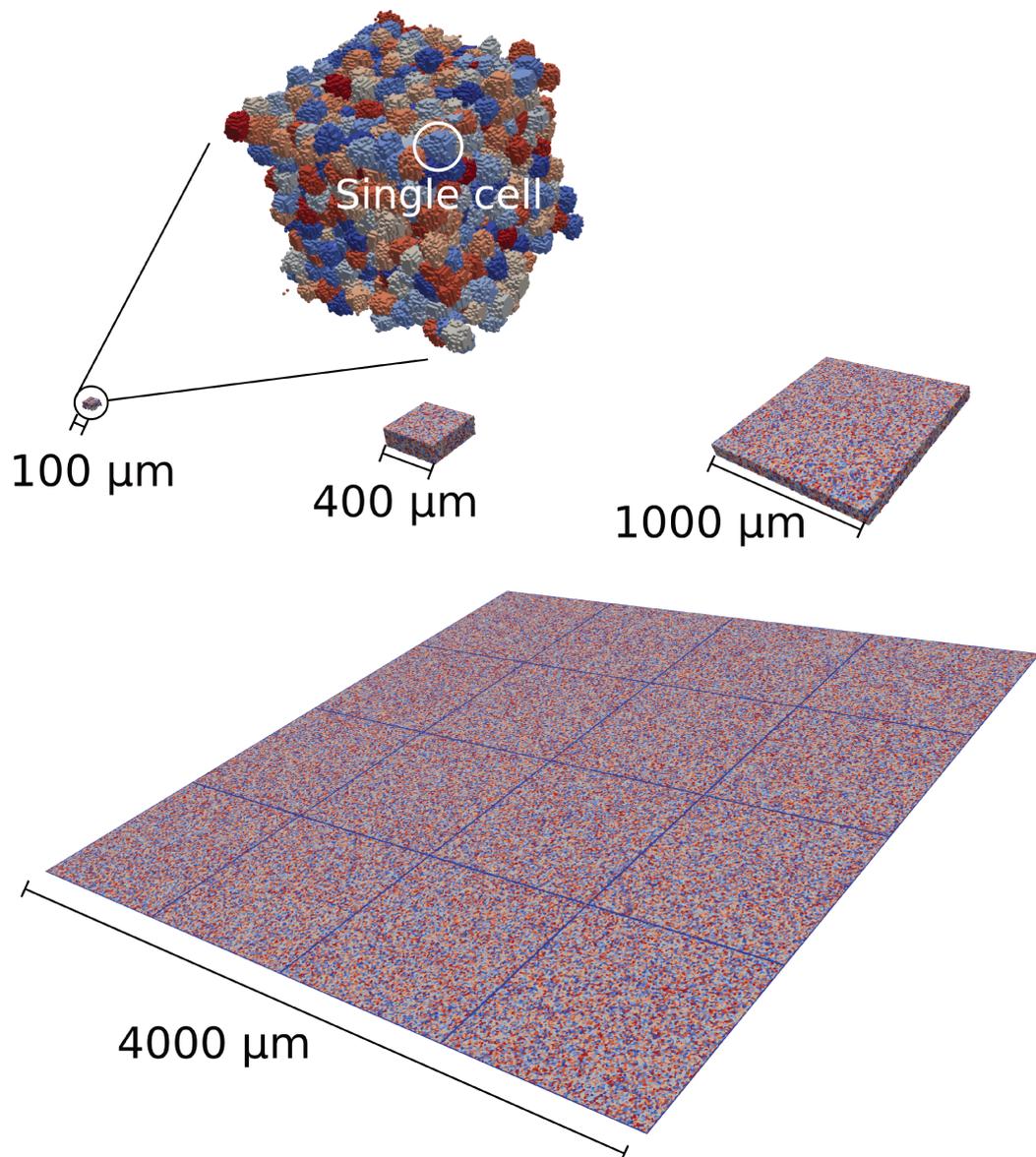
**Macroscale ABM layer and interaction between the layers** Finally, on the top layer of CiS, information from the lower layers is collected, its impact is calculated, and the cell agent properties are updated. The properties of a single cell agent within CiS are the following:

- ID
- Type
- Volume
- Surface
- Time of birth
- Signal content
- Cell center
- Motility direction

These properties usually change over the course of a simulation. The cell motility can either follow a pure random walk or a persistent random walk, and the motility direction is periodically updated accordingly. The volume, surface and center values are updated as the CPM lattice evolves. The signal values change depending on the mesoscale interactions. In addition to changing the agent properties, the ABM layer also governs the effect of these properties. Cell division and cell death are implemented here. Cells divide and die depending on customizable conditions, e.g. the concentration of specific signals. When a cell divides, a new agent is added, and half of the CPM lattice points belonging to the dividing cell are changed to the ID of the new agent. In summary, the macroscale layer combines information from the micro- and mesoscale layers and functions as the main point of control. An example of tissue arrangements within systems of multiple sizes is shown in Figure 1.2 (see also Appendices A and B).

### 1.3 Divide and conquer

The long-term goal of the development of CiS is to build a cellular digital twin of a tumor, with which we can more deeply understand tumor growth and investigate new treatment options. In order to reach this goal, the model first needs to be extended to include all relevant effects. Then, it must be parameterized using *in vivo* data. Unfortunately, we currently lack the data required to do so. This is because if we want to ensure that our single-cell resolving tissue model behaves correctly, we require high-resolution data on the dynamics of *in vivo* tissue. Generating such data represents a significant challenge. The experimentally reachable resolution has increased strongly in recent years, and labeled data on static  $\text{cm}^3$ -sized tissues has already been generated at single cell resolution [23]. While preparing for the arrival of dynamics data, we are currently pursuing a "divide and conquer" approach. Here we study multiple *in vitro* sub systems to partially parameterize CiS and gain new insights on necessary model extensions. By utilizing *in vitro* data we gain the advantage of reaching high measurement resolution, at the cost of system size. Eventually, insights from the studied sub systems will be combined with *in vivo* data to scale back up. During the course of this dissertation, I performed studies of two sub systems. Each of them will be highlighted in the next chapter.



**Figure 1.2:** Scaling example of CiS. Shown are parts of four system volumes of increasing size. For larger volumes, only a slice is shown, but each simulated volume was equal to the cube of the respective side length. The lines within the 4000  $\mu\text{m}^3$  image result from the initial placement of the cells, which was done separately for each 1 mm<sup>3</sup> of the overall volume.

# Chapter 2

## Results

The two areas studied during this thesis are the structural environment of tumors, and the nutrient environment of tumors. Both are instrumental in tumor development [98], and in the following I will describe the main results obtained so far. Following this, I will discuss numerical aspects related to the highly parallelized nature of CiS.

### 2.1 Tumor structural environment

Within biological tissues, cells are surrounded by an extracellular matrix (ECM) [99, 9, 98]. This ECM is composed of structural polymers such as collagen, as well as other proteins, and many signal interactions take place between cells and the ECM [98]. The ECM can also provide a barrier against invasion and metastasis of tumors, as demonstrated in melanoma [100] and breast cancer [9, 101]. In this study we turned to data on tumor spheroids grown in collagen media of varying density [9]. The spheroids exhibited behavior similar to a phase transition at increased collagen density. The invasion behavior changed from predominantly single cell invasion at low densities to collective invasion at higher densities. The spheroid cell behavior in the two phases was reminiscent of that of molecules within liquids versus gases. We aimed to reproduce this behavior with CiS by performing simulations of spheroids surrounded by a solid ECM of varying density. One major part of this process was the parameterization, which required a way of comparing simulated and experimental data. For the comparison of spheroids, no fully established metrics existed, so we developed our own. This was done by extracting several structural features from the raw data, comparing the features between the spheroids, and finally combining all individual comparisons into an overall deviation score (ODS). A high value of the ODS between two spheroids indicates significant morphological difference between them. This project has led

to a publication [96], on which I was shared first author. The full manuscript for this can be found in Appendix A.

After developing and using the ODS, we discovered that our model likely lacks some key functionalities to accurately reproduce the experimentally observed spheroid behavior. Specifically, the solid structure which we used to approximate the ECM is likely insufficient (see also section 3). Our group has therefore started working on including a dynamic ECM into CiS, and such studies are ongoing. In parallel to this, I have turned to the next area: the tumor nutrient environment.

## 2.2 Tumor nutrient environment

A tumor's surrounding has strong impact on its growth. If there is insufficient blood supply, it can only grow up to a certain size [102, 40]. Studying this environment in an *in silico* context may shed new light on the detailed mechanisms of induced angiogenesis and invasion. In this chapter I first describe a project about including detailed mouse brain vasculature data into CiS, and then I briefly discuss an attempt to implement tumor-induced angiogenesis within the model.

### 2.2.1 Static vasculature: mouse brain data

Rosenbauer et al. have previously used CiS to study the nutrient dependence of tumor growth by placing a spherical tumor within a fluctuating nutrient environment [103]. However, one major limitation of this study was the fact that the nutrient environment was highly simplified in the form of a radial field with a single minimum precessing around a central axis. In order to study tumors placed in more realistic environments, I have therefore turned to data by Di Giovanna et al [104], who measured the vasculature of an entire mouse brain at capillary resolution. These data were provided in the form of light-sheet microscopy z-stacks. I processed each of these stacks into a 3-dimensional representation, and analyzed them with respect to the blood vessel network topology. After choosing several stacks of interest, I incorporated their 3-dimensional blood vessel network into the CPM layer of CiS in the form of solids. I then performed simulations of tumors seeded in different parts of the vessel network, and investigated the dependence of the tumor growth on the initial surrounding vessel properties. A manuscript for this project was recently submitted for publication. The corresponding preprint was uploaded to bioRxiv [105] and can be found in Appendix B.

The above study has led to new insights about modeling static vasculature in CiS, and the behavior of tumors growing within existing blood vessel networks. The next step is to find a dynamic representation, in order to simulate fully vascularized tumors.

### 2.2.2 Dynamic vasculature: angiogenesis

Angiogenesis denotes the process of new blood vessel growth. This is induced by signaling molecules such as VEGF (vascular endothelial growth factor), which in turn is released by hypoxic, i.e. oxygen deprived cells [42]. New blood vessels grow along the VEGF-gradient towards the hypoxic area. A growing tumor necessarily has a hypoxic center after reaching sufficient volume, and hence VEGF is released, leading to tumor-induced angiogenesis. This, as described in section 1.1, is one of

the hallmarks of cancer. It is also clear that in order to have an accurate model of the progression from microscopic to vascularized macroscopic tumor, tumor-induced angiogenesis must be included. The following outlines an attempt to incorporate dynamic blood vessel growth into the existing machinery of CiS.

### Blood vessel resolution considerations

The first step in implementing a dynamically growing vasculature is to find a non-static representation of a single blood vessel. A question that needed to be answered here was the detail at which the vessels should be modeled. I was mainly interested in their location and growth direction, and therefore decided not to include detailed structural considerations. Instead, I described a blood vessel as a number of connected segments. To utilize CiS as efficiently as possible, I aimed to use the CPM layer for this.

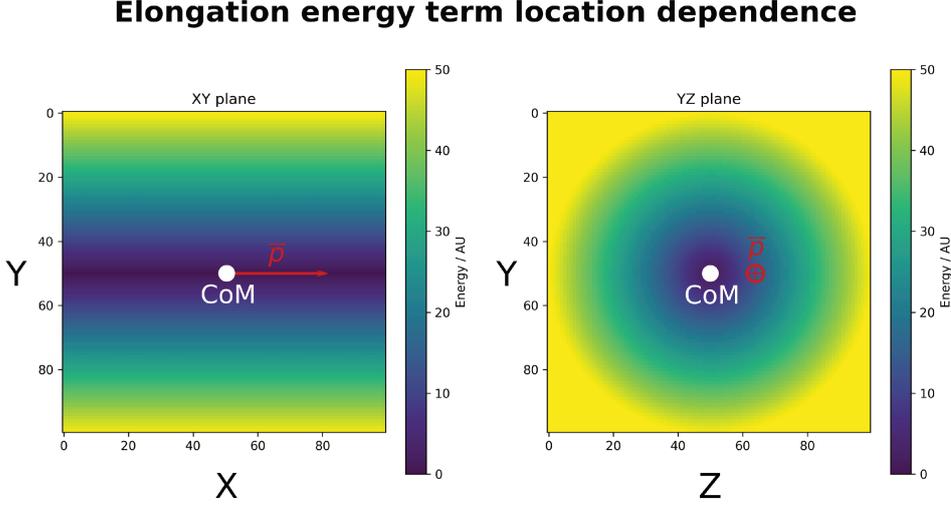
### Elongated cells as blood vessel segments

In the original CPM, a connected region of same ID represents either a cell or the solvent. As described in section 1.2.3, solids can also be included in CiS in the form of static voxels with specific ID. For the inclusion of dynamic blood vessels, I introduced a new type of region: the blood vessel segment (BVS). A BVS functions as a coarse-grained approximation of part of a blood vessel. It combines the lumen and vessel wall into a single object. A full blood vessel is then described by a chain of connected BVSs. Blood vessel growth occurs by utilizing the cell division mechanism of CiS and applying it to BVSs. There are several requirements for this to work. First, a BVS should be elongated compared to regular cells in the CPM. Next, adhesion interactions between two BVSs should only occur between their tips. Thirdly, when a BVS divides, its division plane should be orientated perpendicular to its major axis, such that elongation occurs along this axis. Furthermore, a BVS should only be able to divide if it has less than two neighbors, in order to ensure that only the tip of the blood vessel grows.

**BVS elongation** I achieved the required elongation by incorporating a new "elongation" term into the CPM Hamiltonian:

$$H_{\text{elongation}} = \sum_{i \in \Omega} \lambda_p(\tau(c_i)) \sqrt{\Delta \vec{r}_i^2 - (\Delta \vec{r}_i \cdot \hat{p}(c_i))^2} \quad (2.1)$$

Here  $i$  is the  $i$ -th voxel in the system volume  $\Omega$ ,  $\lambda_p(\tau(c(i)))$  is the coupling factor for the type of the cell to which voxel  $i$  belongs,  $\Delta \vec{r}_i = \vec{r}_i - \vec{r}_{\text{COM}}(c(i))$  is the vector between the  $i$ -th voxel and the center of mass of the cell it belongs to, and  $\hat{p}(c(i))$  is the newly introduced elongation vector of the cell. Including equation 2.1 into



**Figure 2.1:** Energy contribution of the elongation term defined in equation 2.1, depending on the distance from the center of mass  $\vec{r}_{\text{CoM}}(c)$  of cell  $c$ . Here,  $\lambda_p(c) = 1$ ,  $\vec{r}_{\text{CoM}}(c) = (50, 50, 50)$ , and  $\hat{p}(c) = (1, 0, 0)$ . This configuration favors cell voxels close to the axis characterized by  $\vec{r}_{\text{CoM}}(c)$  and  $\hat{p}(c)$ . Left: X-Y plane. Right: Y-Z plane.

the CPM Hamiltonian causes cells with non-zero  $\lambda_p(\tau(c))$  to elongate along their elongation vector. A visualization of the energy term for a single cell is provided in Figure 2.1, and the effect of increased  $\lambda_p(\tau(c))$  is shown in Figure 2.2.

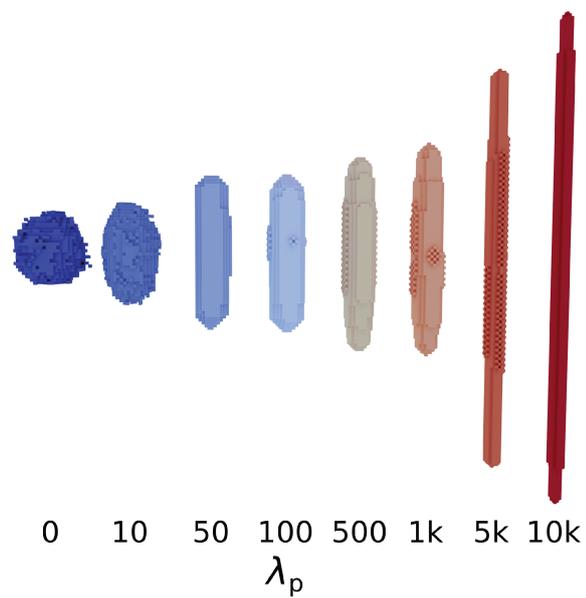
**BVS adhesion** To ensure that adhesion only occurs at the ends of the elongated BVSs, I introduced an additional "tip adhesion" term to the CPM Hamiltonian:

$$H_{\text{tip,adh}} = \sum_{i \in \Omega} \sum_{j \in N(i)} P_{\tau(c_i), \tau(c_j)} \left( \Delta \vec{r}_i \cdot \hat{p}(c_i) + \Delta \vec{r}_j \cdot \hat{p}(c_j) \right) \left( 1 - \delta_{c_i, c_j} \right) \quad (2.2)$$

Here,  $i$  is the  $i$ -th voxel in the system volume  $\Omega$ ,  $j$  is the  $j$ -th neighbor of all neighbors  $N$  of  $i$ ,  $P$  is the adhesion matrix defining the tip adhesion strengths between all cell types,  $\Delta \vec{r}_i = \vec{r}_i - \vec{r}_{\text{CoM}}(c(i))$  is the vector between the  $i$ -th voxel and the center of mass of the cell it belongs to, and  $\delta_{c_i, c_j}$  is the Kronecker delta. Importantly, equation 2.2 includes contributions for both neighbors. Therefore adhesion can occur at orthogonal placement of two BVSs, but is strongest at parallel placement.

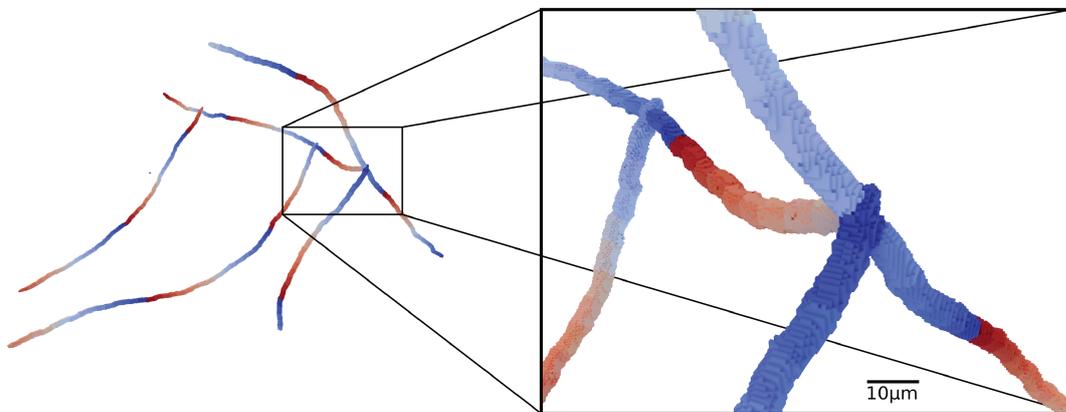
By utilizing the above effects, we can already generate meta-stable networks of connected BVSs. An example of this can be seen in Figure 2.3. In order for such networks to grow, BVS division must be included.

### Elongation energy coupling factor influence



**Figure 2.2:** Cell shape depending on coupling factor  $\lambda_p$  from equation 2.1. Each cell depicted here has the same properties, except for its associated coupling factor. Initially, the behavior is dominated by the volume- and surface terms of the CPM hamiltonian in equation 1.1. As  $\lambda_p$  increases, the cell becomes more elongated.

### Simple vessel network composed of BVSs



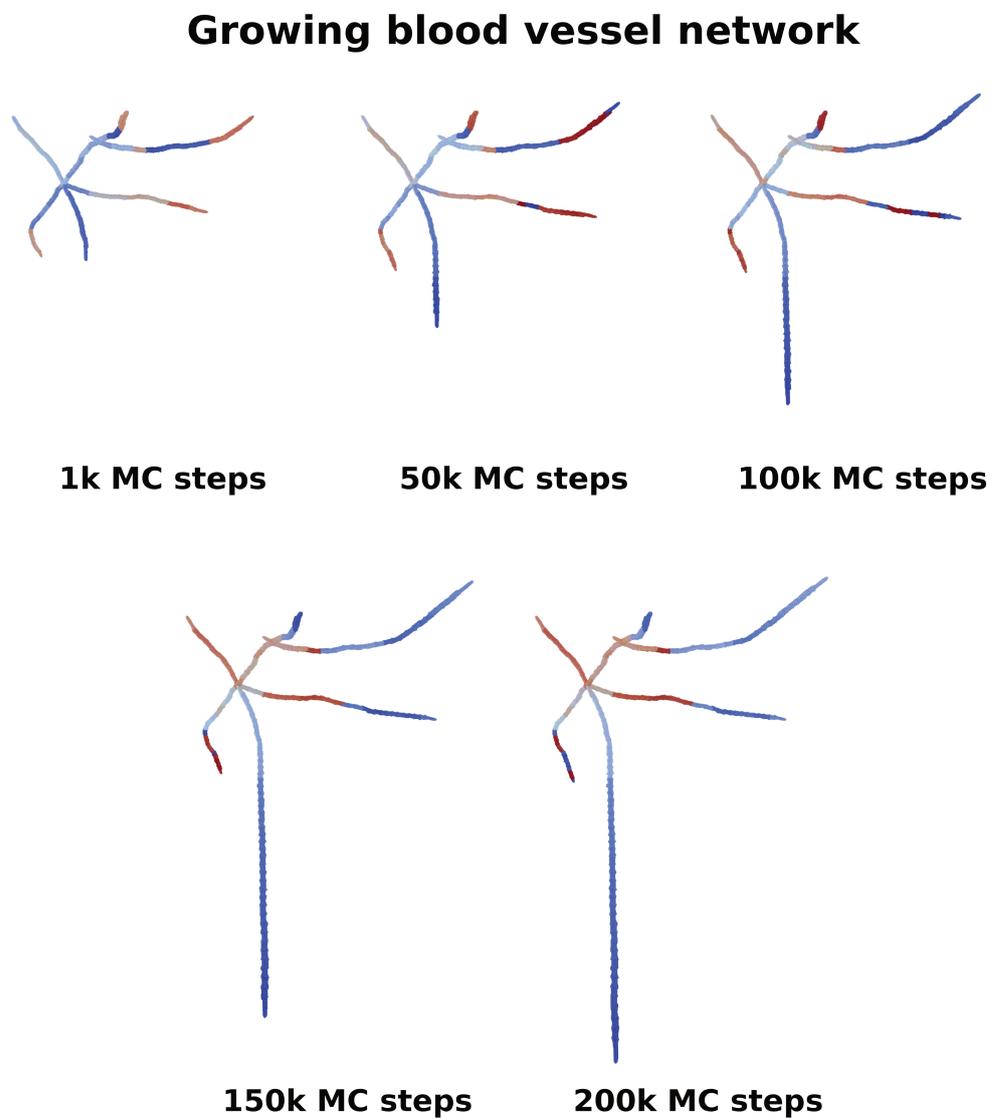
**Figure 2.3:** Example of a vessel network made of BVSs. This network is composed of 213 pre-placed BVSs, each indicated by a color. It contains 4 branches, at which 3-4 BVSs connect. A close-up of one branch is shown on the right.

**BVS division** In CiS, cell division is carried out by first defining a division plane for each dividing cell. Its voxels are then split between the two new cells. The plane is defined by the cell center of mass and a normal vector that is chosen randomly. For BVSs, I have adapted the mechanism to choose  $\hat{p}$  as the normal vector of the division plane. This ensures that the shared surface of old and new BVS is minimized. Finally, to enable division only for those BVSs located at the tip of vessels, the neighbor count  $N_{\text{neighbors},i}$  of each BVS  $i$  is tracked. Division is then only possible for a BVS if  $N_{\text{neighbors},i}$  is less than 2. An example of a growing vessel network is shown in Figure 2.4.

**Gradient-driven growth** The mechanisms described above are already implemented in CiS. However, further machinery is required in order to fully achieve signal-gradient driven vessel extension. Importantly, a mechanism for changing the major axis of a BVS must be included such that it can orient itself towards the signal gradient. This change is not implemented yet, and would need to occur either at the time of cell division or dynamically for each agent.

### Evaluation of this approach

The main motivation for the usage of BVSs was that they could utilize the existing CPM layer within CiS. However, working within this confined space necessarily entails some limitations. First of all, connected BVSs are only metastable, and may break over time, if adhesion parameters are not chosen correctly. Next, the formulation of equation 2.1 does not enforce elongation in a way that mechanically displaces other BVSs. Hence, BVSs often remain confined after division within a growing vessel. Furthermore, the dynamic inclusion of vessel branches remains an unsolved problem. At the current state, it is only possible to "seed" branches in the beginning, from which blood vessels can grow. Finally, BVS thickness is not adjusted, which is another issue for branched structures. In line with this, the formulation does not include the finite rate at which blood flows through the vessels. So far, this is therefore an incomplete formulation. A dynamic blood vessel network may be better achieved by finding a different representation (see section 3).



**Figure 2.4:** Example of a growing blood vessel composed of BVSs. Shown are five snapshots of the vessel network, simulated for a duration of 200 000 MC steps. The network grows along its tips, and retains its alignment.

## 2.3 Numerical aspects

The previous two sections have focused on the biophysical studies performed during my doctoral studies. However, these studies also involved many technical aspects, which are of their own importance and should be mentioned. I first highlight some pitfalls which were discovered in the development of CiS within the HPC context. Then I briefly describe the role of CiS in the JUPITER benchmark, which was utilized to test the components of the JUPITER exascale computer.

### 2.3.1 Pitfalls in HPC development

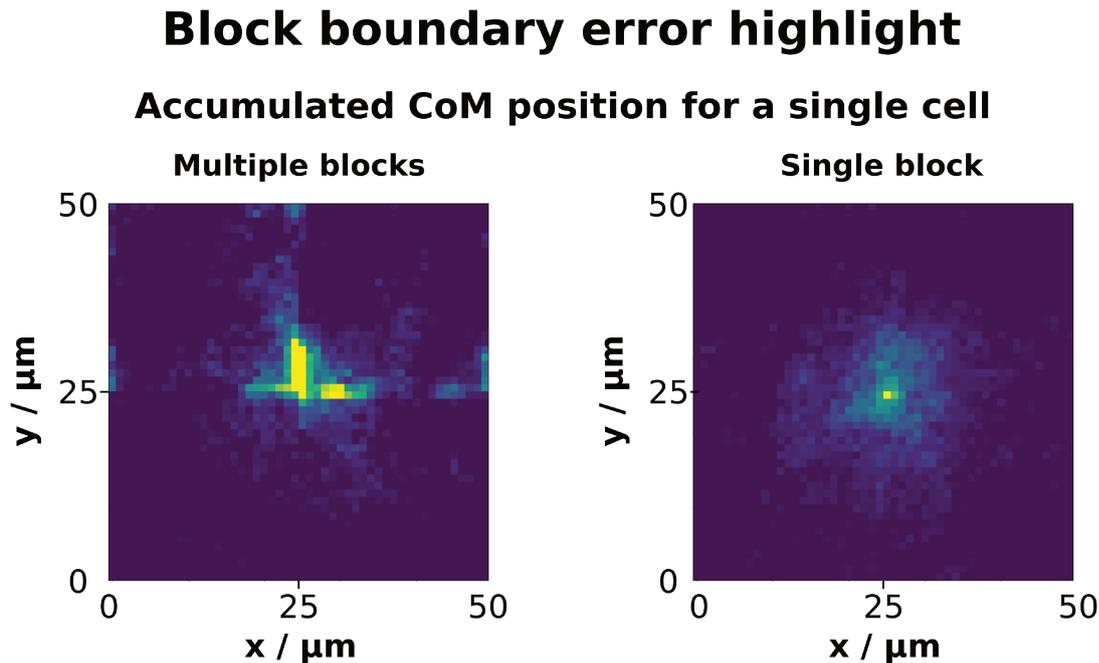
CiS is highly parallelizable, which means that a single simulation can run on thousands of CPU cores. While the information that needs to be exchanged between cores is small, for each new addition to the model the boundary exchanges need to be thoroughly investigated. Additionally, the aspect of simulation output handling, which can be neglected in smaller systems, becomes non-trivial at this level of parallelization. During the course of this project, I found challenges in both of these domains, examples of which are summarized below.

### 2.3.2 Boundary exchange considerations

In order to parallelize the CPM layer of CiS, the volume is divided into a number of sub volumes, known as blocks. Each block is then assigned to a single worker. Since only nearest-neighbor interactions are relevant in the CPM, most of each block can be simulated independently. The exception is the outermost layer, or "halo" of each block, which needs to be exchanged between neighbors.

An information exchange is also required on the agent-based layer. Here, each worker contains its own list of cell agents. Each list contains only those cells which have at least one voxel within the worker's CPM layer block or within a block neighboring it. Importantly, this means that while within a single worker's domain a cell agent is unique, multiple instances of this agent exist in the whole domain. Their properties must be synchronized, such that the properties of block-crossing cell agents, e.g. volume and surface, are correctly updated. It is here that errors such as the following may occur.

**Cell movement around boundaries** Each cell agent  $c$  has a motility vector  $\vec{m}_c$  that determines the direction in which it moves. To implement a random walk, this vector must be periodically shuffled. In CiS, this shuffling takes place only in the block in which the respective cell agent's center of mass is located. This information is then shared with the neighboring blocks. As seen in Figure



**Figure 2.5:** Example of erroneous cell movement due to incorrect boundary exchange. Shown are 2-dimensional histograms of the CoM locations of a single cell over the course of a simulation, once with the simulation parallelized between 8 workers (left), and once with the entire volume simulated by a single worker (right). In both simulations, a single cell was placed in an otherwise empty volume spanning  $50 \cdot 50 \cdot 50 \mu\text{m}$ . In the parallelized case, the cell is found more often close to the block boundaries, which introduces an anisotropy in its random walk movement. The error leading to this behavior has been rectified (see also section 2.3.2).

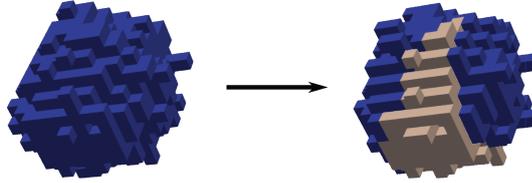
2.5, this can quickly lead to loss of isotropy, if the information exchange is done incorrectly. In the present case, both the center of mass and  $\vec{m}_c$  were incorrectly synchronized. This error has since been fixed.

**Cell division across boundaries** An additional source of error is the choice of voxel coordinates. During cell division, a division plane is defined, and all voxels are assigned to old and new cell via their location relative to this plane. As seen in Figure 2.6, global voxel coordinates must be used here, since otherwise the distribution of voxels is incorrect after division.

### 2.3.3 Parallel IO

CiS has two main types of output. In the first, all agent properties are written out into a `.csv` file, and in the latter the CPM volume is written out into a `.vti` file. At the beginning of my project, both output types already contained parallelized

### Cell division block boundary error



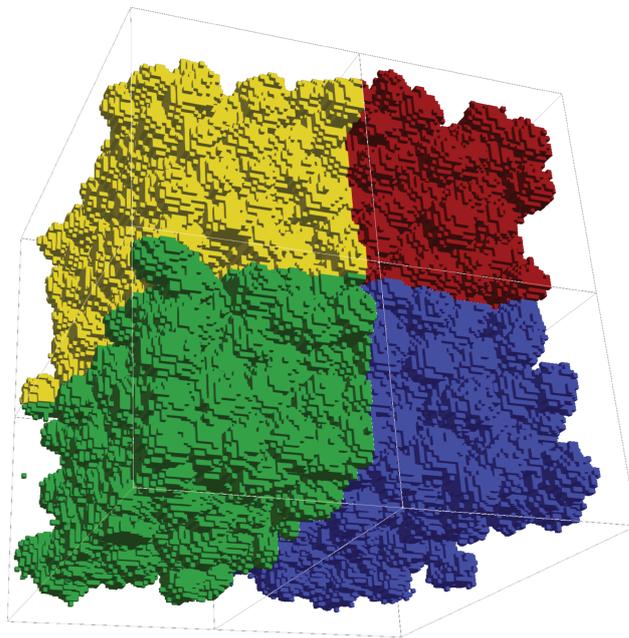
**Figure 2.6:** Example of erroneous cell division across block boundaries. Shown is a single cell located at a block boundary immediately before division (left), and immediately after (right). Due to an inconsistency between using relative and global voxel coordinates, voxels are wrongly distributed between old and new cell. The error leading to this behavior has been rectified (see also section 2.3.2)

versions, in which each worker writes into the same file instead of one file per worker. However, I found that at very high worker number (e.g. 38 400 for the  $4000\ \mu\text{m}^3$  system shown in Figure 1.2) the CPM volume writer overloaded the network of JUWELS. Because the alternative of writing thousands individual files was also undesirable, I therefore implemented an intermediate solution. Now, a group size can be defined, and from this, sub groups of workers are determined which write into their own file. For the  $4000\ \mu\text{m}^3$  system shown in Figure 1.2, the group size was set to 1600, and therefore 24 files were written. An additional example is shown in Figure 2.7, in which each color represents a single file and two blocks.

### 2.3.4 JUPITER benchmark

HPC is an evolving field, and the hardware of a newly built supercomputer may quickly be surpassed by newly developed components. Hence, HPC infrastructure must be updated regularly. At the Jülich Supercomputing Centre (JSC) this already occurred several times, when JUGENE was surpassed by JUQUEEN in 2012, which was in turn surpassed by JUWELS in 2020. Currently, the construction of JUPITER, one of the first European exascale machines, is underway. To ensure the requirement of the best possible hardware for its construction, a benchmark was designed. It included applications from many research groups at JSC, among them CiS under the name "nastja". For the benchmark, a specific use-case (adhesion-driven cell sorting [106]) was chosen, and the system was parallelized at increasing number of cores. The scaling behavior on the new hardware was then analyzed. The development of the benchmark resulted in a publication at the SC24 conference [107]. Its preprint can be found in appendix C.

### Division of simulation volume into multiple sub volumes



**Figure 2.7:** Visualization of the splitting of simulation output into multiple files. Each color indicates a single sub file of the entire volume. The volume was split into 8 blocks, and the group size of the `.vti` output was 2 (see section 2.3.3).

# Chapter 3

## Conclusion and outlook

Cancer is a complex set of diseases, each with unique causes and challenges. This makes the development of universal treatments difficult. To combat these challenges, scientists employ a multi-faceted approach involving experimental and computational methods to study and combat cancer. Experimentally, researchers analyze individual cells using advanced sequencing and motility tracking methods [7, 8], and study the behavior of cell aggregates in *in vitro* tissue models like tumor spheroids and organoids [9, 10]. *In vivo* treatments are also being further developed, thereby enabling new insights [11]. Computational approaches, such as ML-based analyses and mechanistic models like digital twins, complement experimental efforts.

The concept of the digital twin has already been used in multiple fields, such as aerospace engineering [108], urban planning [109], and AI training [110]. Within those fields, tremendous success was achieved by fully representing the system of study in digital form. To replicate such success, the concept is emerging in other fields as well. In Biology, digital twins are very promising candidates for furthering the development of personalized medicine. By investigating treatment options within an *in silico* context, treatments can be ruled out without causing harm to the patient. The tailoring of such treatments to the individual is particularly useful in cancer therapy, because cancer is such a heterogeneous disease whose progression is highly variable depending on the surrounding conditions. Therefore, digital twins promise to augment existing medicine, which still focuses on low resolution data and costly *in vitro* systems [111].

Currently, no true digital twins of tumors exist. It is clear from the scale-bridging nature of cancer, that a working digital twin must be able to simulate tissues which resolve individual cells. The technology for simulating such cell aggregates containing millions of cells is already available [17, 81], and modern super

computing infrastructures allow for handling their tremendous computational cost. On the other hand, the models capable of simulating tissue at the needed resolution still require further tailoring and parameterization [17]. The challenge in tailoring them is to determine, which biological effects need to be incorporated into the models to fully capture tumor behavior, and at which detail these effects need to be simulated. The amount of data available for the subsequent parameterization is growing [23], but it will likely take several more years until this is ready.

The goal of this project was to investigate the requirements and pitfalls of simulating a digital twin of an *in vivo* tumor, and to continue the development of a model capable of doing so. The CiS tissue simulation framework already fulfils multiple key criteria in this regard. The most important of these criteria is its capability of simulating millions of cells at subcellular resolution. In addition to this, the behavior of the individual cells can be widely customized, enabling the capture of highly heterogeneous tissues. However, due to the scarcity of high-resolution *in vivo* data on tumors, CiS cannot be easily parameterized. Hence, during this project, I have employed a "divide and conquer" approach. This means that CiS was used in multiple studies to investigate sub systems for which high-resolution *in vitro* data were available. The systems I analyzed concerned both the structural and nutrient environments of tumors. During these studies I extended the model where it was required:

- I added an ECM degradation mechanism in order to simulate the invasion behavior of MDA-MB-231 breast cancer spheroids [96]. This was previously not possible utilizing the existing machinery of CiS, in which solid ECM could be placed, but simply acted as a barrier or adhesion surface for cells. By implementing a stochastic method in which cells were capable of removing adjacent ECM voxels, I was able to simulate the detachment of individual cells from the main spheroid bulk, a behavior observed experimentally [9].
- I implemented a pipeline for loading 3-dimensional blood vessel data [104] into the CPM layer of CiS. This led to a strong improvement of the nutrient landscape within our simulations, which previously had to rely on a simplified representation [103]. After performing several simulations within different parts of the mouse brain, I was able to confirm the strong impact of vessel density on tumor growth [105].
- I explored a novel way of implementing dynamic vasculature within the CPM layer of CiS. After defining the blood vessel segment (BVS), I implemented a new elongation term into the CPM hamiltonian, added cell tip-adhesion, and neighbor-dependent division. While the goal of a functional dynamic

vasculature was not reached, the implemented BVS-associated effects may also provide useful for cell simulation.

- Finally, I dealt with multiple issues related to the highly parallelized nature of CiS. This included multiple optimizations of the block boundary exchange functionality of CiS, in which I fixed several errors. Furthermore, I improved the parallel IO pipeline for handling large CPU node counts. Simulations involving thousands of individual CPU cores would previously overload the network during simulation output writing. I rectified this by dividing the workload into sub groups of CPU cores which wrote output into individual files. Through this, I was able to simulate systems of  $4 \cdot 4 \cdot 4 \text{ mm}^3$  at  $\mu\text{m}$  resolution.

The study of these multiple projects has led to new insight into tumor simulations. During the investigation of spheroids, we developed the overall deviation score (ODS), a metric to quantitatively compare tumor spheroids. This metric, based on individual cell properties and spheroid bulk properties, enabled us to compare our simulated spheroids to experimental data provided by our collaborators [96]. It was also used in a project within my group whose goal was to find the optimal objective function for the parameterization of our model. While working with the spheroid data, we further discovered that an extension of CiS to include dynamic ECM is necessary to fully capture the behavior of invading cells. It is known that such cells do not only degrade the ECM but actively remodel it [112], and we found that this behavior cannot be neglected when trying to reproduce the experimental data. Further studies to do so with an ECM representation relying on elastic lattice methods are ongoing within my group.

During my studies of the nutrient environment, I performed an analysis of the blood vessel structure within detailed vasculature data of a mouse brain [104]. I was able to convert these data into a 3-dimensional representation and to include them into CiS. After performing tumor growth simulations in a multitude of representative sub parts of the mouse brain, I was able to show that tumor growth is mainly driven by the blood vessel density, and not the network length.

There is much more that can and needs to be done on the path to a full digital twin of tumor growth. First of all, as mentioned before, tumors only induce vascularization after reaching multiple mm in size [102, 40]. Hence, simulations only become truly relevant for therapy development once they can include this scale. With the included and validated blood vessel data [104] we can now tap into a nutrient distribution network spanning multiple  $\text{cm}^3$  of tissue. Using this, we can begin to study the growth of tumors in a clinically relevant system size.

Hence, future projects will attempt to do so and further utilize the richness of the data by di Giovanna et al. [104]. One obvious obstacle that still needs to be overcome during these studies is the formulation of dynamic vasculature. The strategy attempted in section 2.2.2 requires more work to be applicable to a fully dynamic network. A possibility to use the existing mechanisms would be to augment a static vasculature built using the vasculature data by Di Giovanna et al. [104]. BVSs could be seeded along this static vasculature at the start of a simulation. Their growth could then be mediated by a signal secreted by hypoxic tumor cells, thereby mimicking VEGF-gradient driven growth. This is untested, however, and requires further study before its efficacy can be determined. To reach a fully dynamic network, the strategy likely must be re-thought more extensively. An existing approach by Shirinifard et al., which also utilizes the CPM, may be suitable [14], but still needs to be implemented and evaluated.

An aspect that has increasingly come into focus during my work with dynamic vasculature is that of signal and nutrient transport. As discussed in section 1.2.3, this is currently implemented as a contact-driven diffusion process. However, on the one hand this only works in tightly packed tissues, and on the other hand it limits the resolution of the diffusion process to the volume of the individual cells. A more promising approach would be the addition of a dedicated diffusion layer that includes its own field. This would be a second discrete lattice on top of the CPM layer, in which signals would diffuse freely. This has not been implemented so far to keep computational cost down. However, with the advent of GPUs of steadily increasing computational power, it is increasingly feasible to add such a layer. Since GPUs are suited for numerically solving PDEs, they are great candidates for such a task. Furthermore, newer supercomputers increasingly focus more on GPUs than on CPUs, and therefore porting CiS to GPU usage is an attractive prospect.

In terms of simulating treatment, simplified chemotherapy and radiotherapy treatment mechanisms already exist within CiS [113]. However, using only these does not fully reflect the possibilities within cancer therapy, as this also includes others, such as immunotherapy [114] and anti-angiogenic therapy [42]. Therefore, additional work needs to be done to include such therapy options into CiS. In order to incorporate immunotherapy, the role of the immune system has to be included into CiS, an aspect that has not been addressed yet. As discussed in section 1.1, this plays a huge role in the development of tumors, and the tumor microenvironment is usually filled with immune cells. Future work on CiS needs to incorporate an acceptable representation of this immune action. A concern here is the metabolic detail at which the individual cells are currently simulated. Cells contain complex

metabolic and signaling pathways [115], and since these are not captured by CiS, this is likely currently not sufficient to reproduce the wide range of different cell types and behaviors observed *in vitro* and *in vivo*. Before extending the model, immune cells must be studied, their required characteristics and capabilities need to be defined, and data must be found on which to test the function of the simulated cells. One obvious effect that needs to be investigated is the capability of T-cells to seek and neutralize cancerous cells.

Tumor stem cells are another aspect requiring further study. It is known that tumor stem cells contribute to the chemotherapy resistance of tumors [116], and thereby influence tumor robustness. Previously, Rosenbauer et al. have included tumor stem cells within CiS [113], but this was done in a qualitative fashion. To capture their true behavior, data is once again required.

Finally, it is clear that to truly parameterize CiS, data from many different sources are required. Hence, it will be necessary to build a generalized framework for data integration. Such a framework will need to be capable of handling many different types of data, and of associating them to relevant model parameters. Once this exists, the actual process of parameterization will bring further complexity due to the computational cost of individual simulations. Smaller scale models can rely on repeated use of parameter fitting algorithms which iteratively vary parameters to maximize an objective function. At larger scale, the rising computational cost prohibits this, and therefore specialized algorithms requiring fewer iterations are necessary.

The field of cancer research is far from exhausted. While our understanding is constantly growing, new questions and challenges are simultaneously appearing. On the experimental side, obtaining high resolution, time-resolved data of *in vivo* tissue would be highly valuable, although this represents an undeniably challenging problem. On the computational side, there is much more to do until reliable digital twins of tumor growth are applicable and can be utilized to improve personalized medicine. At the same time, the benefits of a working digital twin emphasize the merit of the scientific effort in this field. Furthermore, the advent of exascale computing underlines the fact that the technical resources necessary for such models are increasingly available. Hence, the conditions for the further development of digital twins are excellent. The work I performed during my dissertation has moved CiS slightly closer on the road towards a fully realized digital twin, and future work will continue on this path even further.

# Bibliography

- [1] X. Ma and H. Yu, “Global burden of cancer.,” *The Yale journal of biology and medicine*, vol. 79, pp. 85–94, Dec 2006.
- [2] L. Hanin, “Why victory in the war on cancer remains elusive: biomedical hypotheses and mathematical models.,” *Cancers*, vol. 3, pp. 340–67, Jan 2011.
- [3] A. Upadhyay, “Cancer: An unknown territory; rethinking before going ahead.,” *Genes & diseases*, vol. 8, pp. 655–661, Sep 2021.
- [4] D. Hanahan and R. A. Weinberg, “The hallmarks of cancer,” *Cell*, vol. 100, pp. 57–70, Jan. 2000.
- [5] D. Hanahan and R. A. Weinberg, “Hallmarks of cancer: the next generation.,” *Cell*, vol. 144, pp. 646–74, Mar 2011.
- [6] D. Hanahan, “Hallmarks of Cancer: New Dimensions,” *Cancer Discovery*, vol. 12, pp. 31–46, 01 2022.
- [7] Y. Lei, R. Tang, J. Xu, W. Wang, B. Zhang, J. Liu, X. Yu, and S. Shi, “Applications of single-cell sequencing in cancer research: progress and perspectives,” *Journal of Hematology & Oncology*, vol. 14, p. 91, June 2021.
- [8] A. Nousi, M. T. Sogaard, M. Audoin, and L. Jauffred, “Single-cell tracking reveals super-spreading brain cancer cells with high persistence,” *Biochemistry and Biophysics Reports*, vol. 28, p. 101120, 2021.
- [9] W. Kang, J. Ferruzzi, C.-P. Spatarelu, Y. L. Han, Y. Sharma, S. A. Koehler, J. A. Mitchel, A. Khan, J. P. Butler, D. Roblyer, M. H. Zaman, J.-A. Park, M. Guo, Z. Chen, A. F. Pegoraro, and J. J. Fredberg, “A novel jamming phase diagram links tumor invasion to non-equilibrium phase separation.,” *iScience*, vol. 24, p. 103252, Nov 2021.

- [10] Y. Zhang, M. Liu, N. Xie, Z. Wang, C. Yu, J. Li, and X. Zhou, “Cancer research revolutionized: Unveiling the power of organoids and their therapeutic potential in oncology,” *hLife*, 2024.
- [11] X. Liu, H. Jiang, and X. Wang, “Advances in cancer research: Current and future diagnostic and therapeutic strategies,” *Biosensors*, vol. 14, no. 2, 2024.
- [12] B. Kinnersley, A. Sud, A. Everall, A. J. Cornish, D. Chubb, R. Culliford, A. J. Gruber, A. Lärkeryd, C. Mitsopoulos, D. Wedge, and R. Houlston, “Analysis of 10,478 cancer genomes identifies candidate driver genes and opportunities for precision oncology,” vol. 56, no. 9, pp. 1868–1877. Publisher: Nature Publishing Group.
- [13] L. Lopez-Perez, E. Georga, C. Conti, V. Vicente, R. García, L. Pecchia, D. Fotiadis, L. Licitra, M. F. Cabrera, M. T. Arredondo, and G. Fico, “Statistical and machine learning methods for cancer research and clinical practice: A systematic review,” vol. 92, p. 106067.
- [14] A. Shirinifard, J. S. Gens, B. L. Zaitlen, N. J. Popławski, M. Swat, and J. A. Glazier, “3d multi-cell simulation of tumor growth and angiogenesis,” *PLOS ONE*, vol. 4, pp. 1–11, 10 2009.
- [15] A. Szabó and R. M. H. Merks, “Cellular potts modeling of tumor growth, tumor invasion, and tumor evolution,” *Frontiers in Oncology*, vol. 3, 2013.
- [16] J. Metzcar, Y. Wang, R. Heiland, and P. Macklin, “A review of cell-based computational modeling in cancer biology,” *JCO Clinical Cancer Informatics*, no. 3, pp. 1–13, 2019. PMID: 30715927.
- [17] A. Montagud, M. P. de Leon, and A. Valencia, “Systems biology at the giga-scale: Large multiscale models of complex, heterogeneous multicellular systems,” *Current Opinion in Systems Biology*, vol. 28, p. 100385, 2021.
- [18] R. Laubenbacher, F. Adler, G. An, F. Castiglione, S. Eubank, L. L. Fonseca, J. Glazier, T. Helikar, M. Jett-Tilton, D. Kirschner, P. Macklin, B. Mehrad, B. Moore, V. Pasour, I. Shmulevich, A. Smith, I. Voigt, T. E. Yankeelov, and T. Ziemssen, “Toward mechanistic medical digital twins: some use cases in immunology,” *Frontiers in Digital Health*, vol. 6, 2024.
- [19] S. Runser, R. Vetter, and D. Iber, “Simucell3d: three-dimensional simulation of tissue mechanics with cell polarization,” *Nature Computational Science*, vol. 4, pp. 299–309, Apr 2024.

- [20] J. A. Gallaher, S. C. Massey, A. Hawkins-Daarud, S. S. Noticewala, R. C. Rockne, S. K. Johnston, L. Gonzalez-Cuyar, J. Juliano, O. Gil, K. R. Swanson, P. Canoll, and A. R. A. Anderson, “From cells to tissue: How cell scale heterogeneity impacts glioblastoma growth and treatment response.,” *PLoS computational biology*, vol. 16, p. e1007672, Feb 2020.
- [21] N. Galldiks, P. Lohmann, N. L. Albert, J. C. Tonn, and K.-J. Langen, “Current status of pet imaging in neuro-oncology.,” *Neuro-oncology advances*, vol. 1, p. vdz010, May-Dec 2019.
- [22] Y. Han, Y. Li, W. E. Wu, and Z. Liu, “Noninvasive strategies of cell-tracking in vivo,” *TrAC Trends in Analytical Chemistry*, vol. 172, p. 117616, 2024.
- [23] A. L. Kiemen, A. M. Braxton, M. P. Grahn, K. S. Han, J. M. Babu, R. Reichel, A. C. Jiang, B. Kim, J. Hsu, F. Amoa, S. Reddy, S.-M. Hong, T. C. Cornish, E. D. Thompson, P. Huang, L. D. Wood, R. H. Hruban, D. Wirtz, and P.-H. Wu, “Coda: quantitative 3d reconstruction of large tissues at cellular resolution,” *Nature Methods*, vol. 19, pp. 1490–1499, Nov. 2022.
- [24] F. Cornelis, O. Saut, P. Cumsille, D. Lombardi, A. Iollo, J. Palussiere, and T. Colin, “In vivo mathematical modeling of tumor growth from imaging data: Soon to come in the future?,” *Diagnostic and Interventional Imaging*, vol. 94, no. 6, pp. 593–600, 2013.
- [25] J. S. Brown, S. R. Amend, R. H. Austin, R. A. Gatenby, E. U. Hammarlund, and K. J. Pienta, “Updating the definition of cancer.,” *Molecular cancer research : MCR*, vol. 21, pp. 1142–1147, Nov 2023.
- [26] C. Mattiuzzi and G. Lippi, “Current cancer epidemiology.,” *Journal of epidemiology and global health*, vol. 9, pp. 217–222, Dec 2019.
- [27] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell. 4th edition*. New York: Garland Science, 2002.
- [28] M. Benary, S. Bohn, M. Lüthen, I. K. Nolis, N. Blüthgen, and A. Loewer, “Disentangling pro-mitotic signaling during cell cycle progression using time-resolved single-cell imaging,” *Cell Reports*, vol. 31, Apr. 2020.
- [29] S. Sinha, A. Farfel, K. E. Luker, B. A. Parker, K. T. Yeung, G. D. Luker, and P. Ghosh, “Growth signaling autonomy in circulating tumor cells aids metastatic seeding.,” *PNAS nexus*, vol. 3, p. pgae014, Feb 2024.
- [30] S. A. Aaronson, “Growth factors and cancer.,” *Science (New York, N.Y.)*, vol. 254, pp. 1146–53, Nov 1991.

- [31] A. R. M. R. Amin, P. A. Karpowicz, T. E. Carey, J. Arbiser, R. Nahta, Z. G. Chen, J.-T. Dong, O. Kucuk, G. N. Khan, G. S. Huang, S. Mi, H.-Y. Lee, J. Reichrath, K. Honoki, A. G. Georgakilas, A. Amedei, A. Amin, B. Helderich, C. S. Boosani, M. R. Ciriolo, S. Chen, S. I. Mohammed, A. S. Azmi, W. N. Keith, D. Bhakta, D. Halicka, E. Niccolai, H. Fujii, K. Aquilano, S. S. Ashraf, S. Nowsheen, X. Yang, A. Bilsland, and D. M. Shin, “Evasion of anti-growth signaling: A key step in tumorigenesis and potential target for treatment and prophylaxis by natural compounds.,” *Seminars in cancer biology*, vol. 35 Suppl, pp. S55–S77, Dec 2015.
- [32] R. A. Weinberg, “The retinoblastoma protein and cell cycle control,” *Cell*, vol. 81, no. 3, pp. 323–330, 1995.
- [33] S. Elmore, “Apoptosis: a review of programmed cell death.,” *Toxicologic pathology*, vol. 35, pp. 495–516, Jun 2007.
- [34] R. S. Y. Wong, “Apoptosis in cancer: from pathogenesis to treatment,” *Journal of Experimental & Clinical Cancer Research*, vol. 30, p. 87, Sept. 2011.
- [35] S. Poliwoda, N. Noor, E. Downs, A. Schaaf, A. Cantwell, L. Ganti, A. D. Kaye, L. I. Mosel, C. B. Carroll, O. Viswanath, and I. Urits, “Stem cells: a comprehensive review of origins and emerging clinical roles in medical practice.,” *Orthopedic reviews*, vol. 14, p. 37498, 2022.
- [36] P. Sapienza and F. A. Mallette, “Cellular senescence in postmitotic cells: Beyond growth arrest,” *Trends in Cell Biology*, vol. 28, no. 8, pp. 595–607, 2018.
- [37] R. Di Micco, V. Krizhanovsky, D. Baker, and F. d’Adda di Fagagna, “Cellular senescence in ageing: from mechanisms to therapeutic opportunities,” *Nature Reviews Molecular Cell Biology*, vol. 22, pp. 75–95, Feb. 2021.
- [38] L. Kelland, “Targeting the Limitless Replicative Potential of Cancer: The Telomerase/Telomere Pathway,” *Clinical Cancer Research*, vol. 13, pp. 4960–4963, 09 2007.
- [39] S. C. Akıncılar, J. Y. H. Chua, Q. F. Ng, C. H. T. Chan, Z. Eslami-S, K. Chen, J.-L. Low, S. Arumugam, L. Aswad, C. Chua, I. B. Tan, R. Das-Gupta, M. J. Fullwood, and V. Tergaonkar, “Identification of mechanism of cancer-cell-specific reactivation of htert offers therapeutic opportunities for blocking telomerase specifically in human colorectal cancer.,” *Nucleic acids research*, vol. 51, pp. 1–16, Jan 2023.

- [40] M. K. Gupta and R.-Y. Qin, "Mechanism and its regulation of tumor-induced angiogenesis.," *World journal of gastroenterology*, vol. 9, pp. 1144–55, Jun 2003.
- [41] R. Lugano, M. Ramachandran, and A. Dimberg, "Tumor angiogenesis: causes, consequences, challenges and opportunities.," *Cellular and molecular life sciences : CMLS*, vol. 77, pp. 1745–1770, May 2020.
- [42] Z.-L. Liu, H.-H. Chen, L.-L. Zheng, L.-P. Sun, and L. Shi, "Angiogenic signaling pathways and anti-angiogenic therapy for cancer," *Signal Transduction and Targeted Therapy*, vol. 8, p. 198, May 2023.
- [43] W. Jiang, A. Sanders, M. Katoh, H. Ungefroren, F. Gieseler, M. Prince, S. Thompson, M. Zollo, D. Spano, P. Dhawan, D. Sliva, P. Subbarayan, M. Sarkar, K. Honoki, H. Fujii, A. Georgakilas, A. Amedei, E. Nicolai, A. Amin, S. Ashraf, L. Ye, W. Helferich, X. Yang, C. Boosani, G. Guha, M. Ciriolo, K. Aquilano, S. Chen, A. Azmi, W. Keith, A. Bilsland, D. Bhakta, D. Halicka, S. Nowsheen, F. Pantano, and D. Santini, "Tissue invasion and metastasis: Molecular, biological and clinical perspectives," *Seminars in Cancer Biology*, vol. 35, pp. S244–S275, 2015. A broad-spectrum integrative design for cancer prevention and therapy.
- [44] G. P. Gupta and J. Massagué, "Cancer metastasis: Building a framework," *Cell*, vol. 127, no. 4, pp. 679–695, 2006.
- [45] D. R. Bielenberg and B. R. Zetter, "The contribution of angiogenesis to the process of metastasis.," *Cancer journal (Sudbury, Mass.)*, vol. 21, pp. 267–73, Jul-Aug 2015.
- [46] A. Labani-Motlagh, M. Ashja-Mahdavi, and A. Loskog, "The tumor microenvironment: A milieu hindering and obstructing antitumor immune responses.," *Frontiers in immunology*, vol. 11, p. 940, 2020.
- [47] H.-X. Li, S.-Q. Wang, Z.-X. Lian, S.-L. Deng, and K. Yu, "Relationship between tumor infiltrating immune cells and tumor metastasis and its prognostic value in cancer.," *Cells*, vol. 12, Dec 2022.
- [48] L. Zapata, G. Caravagna, M. J. Williams, E. Lakatos, K. AbdulJabbar, B. Werner, D. Chowell, C. James, L. Gourmet, S. Milite, A. Acar, N. Riaz, T. A. Chan, T. A. Graham, and A. Sottoriva, "Immune selection determines tumor antigenicity and influences response to checkpoint inhibitors," *Nature Genetics*, vol. 55, pp. 451–460, Mar. 2023.

- [49] S. K. Kim and S. W. Cho, “The evasion mechanisms of cancer immunity and drug intervention in the tumor microenvironment.,” *Frontiers in pharmacology*, vol. 13, p. 868695, 2022.
- [50] J. Zheng, “Energy metabolism of cancer: Glycolysis versus oxidative phosphorylation (review).,” *Oncology letters*, vol. 4, pp. 1151–1157, Dec 2012.
- [51] M. V. Liberti and J. W. Locasale, “The warburg effect: How does it benefit cancer cells?,” *Trends in biochemical sciences*, vol. 41, pp. 211–218, Mar 2016.
- [52] V. Petrova, M. Annicchiarico-Petruzzelli, G. Melino, and I. Amelio, “The hypoxic tumour microenvironment,” *Oncogenesis*, vol. 7, p. 10, Jan. 2018.
- [53] S. Negrini, V. G. Gorgoulis, and T. D. Halazonetis, “Genomic instability – an evolving hallmark of cancer,” *Nature Reviews Molecular Cell Biology*, vol. 11, pp. 220–228, Mar. 2010.
- [54] A. Mantovani, P. Allavena, A. Sica, and F. Balkwill, “Cancer-related inflammation,” *Nature*, vol. 454, pp. 436–444, July 2008.
- [55] M. Esteller, “Aberrant dna methylation as a cancer-inducing mechanism.,” *Annual review of pharmacology and toxicology*, vol. 45, pp. 629–56, 2005.
- [56] A. V. Paska and P. Hudler, “Aberrant methylation patterns in cancer: a clinical view.,” *Biochemia medica*, vol. 25, pp. 161–76, 2015.
- [57] X. Yu, H. Zhao, R. Wang, Y. Chen, X. Ouyang, W. Li, Y. Sun, and A. Peng, “Cancer epigenetics: from laboratory studies and clinical trials to precision medicine,” *Cell Death Discovery*, vol. 10, p. 28, Jan. 2024.
- [58] E. C. Holland, “Glioblastoma multiforme: the terminator.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 97, pp. 6242–4, Jun 2000.
- [59] A. Claes, A. J. Idema, and P. Wesseling, “Diffuse glioma growth: a guerilla war.,” *Acta neuropathologica*, vol. 114, pp. 443–58, Nov 2007.
- [60] G. P. W. Hordur Mar Kolbeinsson, Sreenivasa Chandana and M. Chung, “Pancreatic cancer: A review of current treatment and novel therapies,” *Journal of Investigative Surgery*, vol. 36, no. 1, p. 2129884, 2023. PMID: 36191926.

- [61] U. Anand, A. Dey, A. K. S. Chandel, R. Sanyal, A. Mishra, D. K. Pandey, V. De Falco, A. Upadhyay, R. Kandimalla, A. Chaudhary, J. K. Dhanjal, S. Dewanjee, J. Vallamkondu, and J. M. Pérez de la Lastra, “Cancer chemotherapy and beyond: Current status, drug candidates, associated risks and progress in targeted therapeutics,” *Genes & Diseases*, vol. 10, no. 4, pp. 1367–1401, 2023.
- [62] G. Chaput and L. Regnier, “Radiotherapy: Clinical pearls for primary care.,” *Canadian family physician Medecin de famille canadien*, vol. 67, pp. 753–757, Oct 2021.
- [63] J. Kleeff and U. Ronellenfitsch, “Surgical oncology: Multidisciplinarity to improve cancer treatment and outcomes.,” in *Current oncology (Toronto, Ont.)*, vol. 28, (Switzerland), pp. 4471–4473, Nov 2021.
- [64] K. A. Amjad MT, Chidharla A, “Cancer chemotherapy [updated 2023 feb 27],” *StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2024 Jan-.* Available from: <https://www.ncbi.nlm.nih.gov/books/NBK564367/>, 2023.
- [65] S. Tohme, R. L. Simmons, and A. Tsung, “Surgery for cancer: A trigger for metastases.,” *Cancer research*, vol. 77, pp. 1548–1552, Apr 2017.
- [66] M. E. Katt, A. L. Placone, A. D. Wong, Z. S. Xu, and P. C. Searson, “In vitro tumor models: Advantages, disadvantages, variables, and selecting the right platform.,” *Frontiers in bioengineering and biotechnology*, vol. 4, p. 12, 2016.
- [67] W. F. Scherer, J. T. Syverton, and G. O. Gey, “STUDIES ON THE PROPAGATION IN VITRO OF POLIOMYELITIS VIRUSES : IV. VIRAL MULTIPLICATION IN A STABLE STRAIN OF HUMAN MALIGNANT EPITHELIAL CELLS (STRAIN HELA) DERIVED FROM AN EPIDERMOID CARCINOMA OF THE CERVIX ,” *Journal of Experimental Medicine*, vol. 97, pp. 695–710, 05 1953.
- [68] R. Rahbari, T. Sheahan, V. Modes, P. Collier, C. Macfarlane, and R. M. Badge, “A novel I1 retrotransposon marker for hela cell line identification.,” tech. rep., England, Apr 2009.
- [69] A. Ertel, A. Verghese, S. W. Byers, M. Ochs, and A. Tozeren, “Pathway-specific differences between tumor cell lines and normal and tumor tissue cells,” *Molecular Cancer*, vol. 5, p. 55, Nov. 2006.

- [70] C. M. Lopes-Ramos, J. N. Paulson, C.-Y. Chen, M. L. Kuijjer, M. Fagny, J. Platig, A. R. Sonawane, D. L. DeMeo, J. Quackenbush, and K. Glass, “Regulatory network changes between cell lines and their tissues of origin.,” *BMC genomics*, vol. 18, p. 723, Sep 2017.
- [71] M. Volleth, M. Zenker, I. Joksic, and T. Liehr, “Long-term culture of ebv-induced human lymphoblastoid cell lines reveals chromosomal instability.,” *The journal of histochemistry and cytochemistry : official journal of the Histochemistry Society*, vol. 68, pp. 239–251, Apr 2020.
- [72] E. Cukierman, R. Pankov, D. R. Stevens, and K. M. Yamada, “Taking cell-matrix adhesions to the third dimension.,” *Science (New York, N.Y.)*, vol. 294, pp. 1708–12, Nov 2001.
- [73] F. Perche and V. P. Torchilin, “Cancer cell spheroids as a model to evaluate chemotherapy protocols,” *Cancer Biology & Therapy*, vol. 13, no. 12, pp. 1205–1213, 2012. PMID: 22892843.
- [74] S. Gunti, A. T. K. Hoke, K. P. Vu, and N. R. J. London, “Organoid and spheroid tumor models: Techniques and applications.,” *Cancers*, vol. 13, Feb 2021.
- [75] A. Al-Ramadan, A. C. Mortensen, J. Carlsson, and M. V. Nestor, “Analysis of radiation effects in two irradiated tumor spheroid models.,” *Oncology letters*, vol. 15, pp. 3008–3016, Mar 2018.
- [76] S. Yang, H. Hu, H. Kung, R. Zou, Y. Dai, Y. Hu, T. Wang, T. Lv, J. Yu, and F. Li, “Organoids: The current status and biomedical applications.,” *MedComm*, vol. 4, p. e274, Jun 2023.
- [77] C. F. Ng and H. B. Frieboes, “Simulation of multispecies desmoplastic cancer growth via a fully adaptive non-linear full multigrid algorithm,” *Frontiers in Physiology*, vol. 9, 2018.
- [78] D. A. Goodin and H. B. Frieboes, “Simulation of 3d centimeter-scale continuum tumor growth at sub-millimeter resolution via distributed computing,” *Computers in Biology and Medicine*, vol. 134, p. 104507, 2021.
- [79] M. Siam, H. Wang, M. K. Lindell, C. Chen, E. I. Vlahogianni, and K. Axhausen, “An interdisciplinary agent-based multimodal wildfire evacuation model: Critical decisions and life safety,” *Transportation Research Part D: Transport and Environment*, vol. 103, p. 103147, 2022.

- [80] I. Vetharanim, W. J. Kelly, G. T. Attwood, and P. J. Harris, “A 3-d model of a perennial ryegrass primary cell wall and its enzymatic degradation,” *Computation*, vol. 2, no. 2, pp. 23–46, 2014.
- [81] M. Berghoff, J. Rosenbauer, F. Hoffmann, and A. Schug, “Cells in silico—introducing a high-performance framework for large-scale tissue modeling,” *BMC bioinformatics*, vol. 21, no. 1, pp. 1–21, 2020.
- [82] M. Gardner, “Mathematical Games - The fantastic combinations of John Conway’s new solitaire game ”life” - M. Gardner - 1970,” *Scientific American*, 1970.
- [83] P. Van Liedekerke, M. M. Palm, N. Jagiella, and D. Drasdo, “Simulating tissue mechanics with agent-based models: concepts, perspectives and some novel results,” *Computational Particle Mechanics*, vol. 2, pp. 401–444, Dec. 2015.
- [84] A. Ghaffarizadeh, R. Heiland, S. H. Friedman, S. M. Mumenthaler, and P. Macklin, “Physicell: An open source physics-based cell simulator for 3-d multicellular systems,” *PLOS Computational Biology*, vol. 14, pp. 1–31, 02 2018.
- [85] A. Nagelkerke, J. Bussink, A. E. Rowan, and P. N. Span, “The mechanical microenvironment in cancer: How physics affects tumours,” *Seminars in Cancer Biology*, vol. 35, pp. 62–70, 2015. Complexity in Cancer Biology.
- [86] F. Aurenhammer, “Voronoi diagrams—a survey of a fundamental geometric data structure,” *ACM Comput. Surv.*, vol. 23, p. 345–405, sep 1991.
- [87] L. Sui, S. Alt, M. Weigert, N. Dye, S. Eaton, F. Jug, E. W. Myers, F. Jülicher, G. Salbreux, and C. Dahmann, “Differential lateral and basal tension drive folding of drosophila wing discs through two distinct mechanisms,” *Nature communications*, vol. 9, p. 4620, Nov 2018.
- [88] D. Bi, X. Yang, M. C. Marchetti, and M. L. Manning, “Motility-driven glass and jamming transitions in biological tissues.,” *Physical review. X*, vol. 6, Apr-Jun 2016.
- [89] F. m. c. Graner and J. A. Glazier, “Simulation of biological cell sorting using a two-dimensional extended potts model,” *Phys. Rev. Lett.*, vol. 69, pp. 2013–2016, Sep 1992.
- [90] F. Y. Wu, “The potts model,” *Rev. Mod. Phys.*, vol. 54, pp. 235–268, Jan 1982.

- [91] E. Ising, “Beitrag zur theorie des ferromagnetismus,” *Zeitschrift für Physik*, vol. 31, pp. 253–258, Feb. 1925.
- [92] N. Guisoni, K. I. Mazzitello, and L. Diambra, “Modeling active cell movement with the potts model,” *Frontiers in Physics*, vol. 6, 2018.
- [93] V. S. Borkar, “Equation of state calculations by fast computing machines,” *Resonance*, vol. 27, pp. 1263 – 1269, 1953.
- [94] M. Berghoff, I. Kondov, and J. Hötzer, “Massively parallel stencil code solver with autonomous adaptive block distribution,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 10, pp. 2282–2296, 2018.
- [95] M. Berghoff and I. Kondov, “Non-collective scalable global network based on local communications,” in *2018 IEEE/ACM 9th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems (scalA)*, pp. 25–32, 2018.
- [96] J. Herold, E. Behle, J. Rosenbauer, J. Ferruzzi, and A. Schug, “Development of a scoring function for comparing simulated and experimental tumor spheroids,” *PLOS Computational Biology*, vol. 19, pp. 1–26, 03 2023.
- [97] P. Müller and A. F. Schier, “Extracellular movement of signaling molecules.,” *Developmental cell*, vol. 21, pp. 145–58, Jul 2011.
- [98] Z. Yuan, Y. Li, S. Zhang, X. Wang, H. Dou, X. Yu, Z. Zhang, S. Yang, and M. Xiao, “Extracellular matrix remodeling in tumor progression and immune escape: from mechanisms to treatments,” *Molecular Cancer*, vol. 22, p. 48, Mar. 2023.
- [99] J. K. Kular, S. Basu, and R. I. Sharma, “The extracellular matrix: Structure, composition, age-related differences, tools for analysis and applications for tissue engineering,” *Journal of Tissue Engineering*, vol. 5, p. 2041731414557112, 2014. PMID: 25610589.
- [100] A. Popovic and S. Tartare-Deckert, “Role of extracellular matrix architecture and signaling in melanoma therapeutic resistance.,” *Frontiers in oncology*, vol. 12, p. 924553, 2022.
- [101] Z. Mai, Y. Lin, P. Lin, X. Zhao, and L. Cui, “Modulating extracellular matrix stiffness: a strategic approach to boost cancer immunotherapy,” *Cell Death & Disease*, vol. 15, p. 307, May 2024.

- [102] M. Chaplain, “Avascular growth, angiogenesis and vascular growth in solid tumours: The mathematical modelling of the stages of tumour development,” *Mathematical and Computer Modelling*, vol. 23, no. 6, pp. 47–87, 1996.
- [103] J. Rosenbauer, M. Berghoff, J. A. Glazier, and A. Schug, “Multiscale modeling of spheroid tumors: Effect of nutrient availability on tumor evolution,” *J. Phys. Chem. B*, vol. 127, pp. 3607–3615, Apr. 2023.
- [104] D. Giovanna, A. Paolo, A. Tibo, L. Silvestri, M. C. Müllenbroich, I. Costantini, A. L. Allegra Mascaro, L. Sacconi, P. Frasconi, and F. S. Pavone, “Whole-brain vasculature reconstruction at the single capillary level,” *Scientific Reports*, vol. 8, p. 12573, Aug. 2018.
- [105] E. Behle, J. Herold, and A. Schug, “Leveraging experimental vasculature data for high resolution brain tumor simulations,” *bioRxiv*, 2024.
- [106] M. S. STEINBERG, “On the mechanism of tissue reconstruction by dissociated cells. i. population kinetics, differential adhesiveness. and the absence of directed migration.” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 48, pp. 1577–82, Sep 1962.
- [107] A. Herten, S. Achilles, D. Alvarez, J. Badwaik, E. Behle, M. Bode, T. Breuer, D. Caviedes-Voullième, M. Cherti, A. Dabah, S. El Sayed, W. Frings, A. Gonzalez-Nicolas, E. B. Gregory, K. H. Mood, T. Hater, J. Jitsev, C. M. John, J. H. Meinke, C. I. Meyer, P. Mezentsev, J.-O. Mirus, S. Nassyr, C. Penke, M. Römmer, U. Sinha, B. von St. Vieth, O. Stein, E. Suarez, D. Willsch, and I. Zhukov, “Application-Driven Exascale: The JUPITER Benchmark Suite ,” in *2024 SC24: International Conference for High Performance Computing, Networking, Storage and Analysis SC*, (Los Alamitos, CA, USA), pp. 468–512, IEEE Computer Society, Nov. 2024.
- [108] A. Ferrari and K. Willcox, “Digital twins in mechanical and aerospace engineering,” *Nature Computational Science*, vol. 4, pp. 178–183, Mar. 2024.
- [109] F. Jiang, L. Ma, T. Broyd, and K. Chen, “Digital twin and its implementations in the civil engineering sector,” *Automation in Construction*, vol. 130, p. 103838, 2021.
- [110] D. Chen and Z. Lv, “Artificial intelligence enabled digital twins for training autonomous cars,” *Internet of Things and Cyber-Physical Systems*, vol. 2, pp. 31–41, 2022.

- [111] R. Augustine, S. N. Kalva, R. Ahmad, A. A. Zahid, S. Hasan, A. Nayeem, L. McClements, and A. Hasan, “3d bioprinted cancer models: Revolutionizing personalized cancer therapy,” *Translational Oncology*, vol. 14, no. 4, p. 101015, 2021.
- [112] J. Ferruzzi, M. Sun, A. Gkousioudi, A. Pilvar, D. Roblyer, Y. Zhang, and M. H. Zaman, “Compressive remodeling alters fluid transport properties of collagen networks - implications for tumor growth,” *Scientific Reports*, vol. 9, p. 17151, Nov. 2019.
- [113] J. Rosenbauer, M. Berghoff, and A. Schug, “Emerging tumor development by simulating single-cell events,” *bioRxiv*, 2020.
- [114] A. D. Waldman, J. M. Fritz, and M. J. Lenardo, “A guide to cancer immunotherapy: from t cell basic science to clinical practice,” vol. 20, no. 11, pp. 651–668. Publisher: Nature Publishing Group.
- [115] J. Su, Y. Song, Z. Zhu, X. Huang, J. Fan, J. Qiao, and F. Mao, “Cell–cell communication: new insights and clinical implications,” vol. 9, no. 1, pp. 1–52. Publisher: Nature Publishing Group.
- [116] N. P. j, S. R. Patil, V. P. Veeraraghavan, S. Daniel, K. R. Aileni, and M. I. Karobari, “Oral cancer stem cells: A comprehensive review of key drivers of treatment resistance and tumor recurrence,” p. 177222.

# Appendices

# Appendix A

## Embedded Publication 1

Julian Herold\*, **Eric Behle\***, Jakob Rosenbauer, Jacopo Ferruzzi, Alexander Schug, *Development of a scoring function for comparing simulated and experimental tumor spheroids*, PLOS Computational Biology, 2023

\*Shared first authors

### Copy permissions

This article was published in an open-access journal with the following copyright remark: *This is an open access article, free of all copyright, and may be freely reproduced, distributed, transmitted, modified, built upon, or otherwise used by anyone for any lawful purpose. The work is made available under the Creative Commons CC0 public domain dedication.*

### Contributions

For this publication, I developed the *Nastjapy* package together with my co-author Julian Herold (50% contribution each). Except for section 4.3, we jointly wrote the entire manuscript. The writing process here was highly collaborative, involving iterative discussions, joint drafting, and refinement. I contributed approximately 50% of the overall writing effort. I extended the code of *Cells in Silico*, and performed all simulations and data analyses. I generated all figures.

### Full article

## RESEARCH ARTICLE

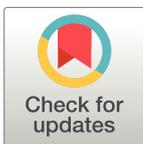
## Development of a scoring function for comparing simulated and experimental tumor spheroids

Julian Herold<sup>1,2</sup>, Eric Behle<sup>3</sup>, Jakob Rosenbauer<sup>3</sup>, Jacopo Ferruzzi<sup>4</sup>, Alexander Schug<sup>3\*</sup>

1 HIDSS4Health - Helmholtz Information and Data Science School for Health, Karlsruhe/Heidelberg, Germany, 2 Steinbuch Centre for Computing, Karlsruhe Institute of Technology, Karlsruhe, Germany, 3 NIC Research Group Computational Structural Biology, Jülich Research Center, Jülich, Germany, 4 Department of Bioengineering, University of Texas at Dallas, Richardson, Texas, United States of America

☞ These authors contributed equally to this work.

\* [al.schug@fz-juelich.de](mailto:al.schug@fz-juelich.de)



## OPEN ACCESS

**Citation:** Herold J, Behle E, Rosenbauer J, Ferruzzi J, Schug A (2023) Development of a scoring function for comparing simulated and experimental tumor spheroids. PLoS Comput Biol 19(3): e1010471. <https://doi.org/10.1371/journal.pcbi.1010471>

**Editor:** Philipp M Altrock, Max Planck Institute for Evolutionary Biology: Max-Planck-Institut für Evolutionsbiologie, GERMANY

**Received:** August 8, 2022

**Accepted:** March 4, 2023

**Published:** March 30, 2023

**Copyright:** This is an open access article, free of all copyright, and may be freely reproduced, distributed, transmitted, modified, built upon, or otherwise used by anyone for any lawful purpose. The work is made available under the [Creative Commons CC0](https://creativecommons.org/licenses/by/4.0/) public domain dedication.

**Data Availability Statement:** The code for the "Cells in Silico" framework is available at <https://gitlab.com/nastja/nastja>. The code for the "Nastjapy" framework is available at <https://gitlab.com/nastja/nastjapy>. Example configuration files for the simulations, as well as the simulation configuration files and raw data are provided at [https://gitlab.com/nastja/nastjapy\\_paper](https://gitlab.com/nastja/nastjapy_paper).

**Funding:** AS, EB, JR This work has been funded by NIC (John von Neumann Institute for Computing)

## Abstract

Progress continues in the field of cancer biology, yet much remains to be unveiled regarding the mechanisms of cancer invasion. In particular, complex biophysical mechanisms enable a tumor to remodel the surrounding extracellular matrix (ECM), allowing cells to invade alone or collectively. Tumor spheroids cultured in collagen represent a simplified, reproducible 3D model system, which is sufficiently complex to recapitulate the evolving organization of cells and interaction with the ECM that occur during invasion. Recent experimental approaches enable high resolution imaging and quantification of the internal structure of invading tumor spheroids. Concurrently, computational modeling enables simulations of complex multicellular aggregates based on first principles. The comparison between real and simulated spheroids represents a way to fully exploit both data sources, but remains a challenge. We hypothesize that comparing any two spheroids requires first the extraction of basic features from the raw data, and second the definition of key metrics to match such features. Here, we present a novel method to compare spatial features of spheroids in 3D. To do so, we define and extract features from spheroid point cloud data, which we simulated using Cells in Silico (CiS), a high-performance framework for large-scale tissue modeling previously developed by us. We then define metrics to compare features between individual spheroids, and combine all metrics into an overall deviation score. Finally, we use our features to compare experimental data on invading spheroids in increasing collagen densities. We propose that our approach represents the basis for defining improved metrics to compare large 3D data sets. Moving forward, this approach will enable the detailed analysis of spheroids of any origin, one application of which is informing *in silico* spheroids based on their *in vitro* counterparts. This will enable both basic and applied researchers to close the loop between modeling and experiments in cancer research.

of the Helmholtz association. AS, JH The present contribution is supported by the Helmholtz Association under the joint research school "HIDSS4Health – Helmholtz Information and Data Science School for Health. JF Collection of experimental data was funded by the National Cancer Institute (NIH grant U01CA202123) and by the Department of Defense (DoD grant W81XWH-15-1-0070). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing interests:** The authors declare that they have no competing interests.

## Author summary

Cells within a tumor use various methods to escape and thereby invade into healthy parts of the body. These methods are studied experimentally by examining tumor spheroids, spherical aggregates of hundreds to thousands of individual cells. Such spheroids can also be simulated, and the comparison of both simulations and experiments is desirable. Here, we present an analysis strategy for the comparison of tumor spheroids of any origin. Using this strategy, we aim to increase the information gained from the data and improve the collaborative potential between experimentalists and theorists.

This is a *PLOS Computational Biology* Methods paper.

## 1 Introduction

The worldwide challenge of fighting cancer is as urgent as ever [1, 2]. When trying to understand the mechanisms driving the disease, one is faced with a complex and wildly inhomogeneous landscape of cellular properties and interactions, which vary both within and between cancer types [3–5]. Furthermore, cancer is not one single disease, but rather refers to a large number of diseases with shared characteristics, which are captured in the hallmarks of cancer [2, 6]. The processes underlying these diseases, such as the rise of malignancy via loss of cell-cell adhesion and subsequent increased motility [7], span a wide range of scales, both in space and in time [8]. To further the understanding of cancer, it is crucial to decipher how these processes interact and lead to the formation of macroscopic invasive tumors. Thus, combating cancer requires input from many different domains of science, such as biology, medicine, and pharmacology, but also physics, computer science, and mathematics [8, 9]. Unfortunately, time-resolved analysis of *in vivo* tumor tissue is challenging, as due to low spatial or temporal resolution of imaging methods, single-cell resolution 4D trajectories are not yet widely applicable. To increase accessibility for analysis, the system has to be divided into smaller subsystems. Thus, *in vitro* and *in silico* models are created, allowing the study of individual aspects of the system. An *in vitro* example is the study of tumor spheroids, which represent a useful model system for studying tumor growth and cell dynamics [10]. Tumor spheroids are spherical arrangements of hundreds to thousands of cells, which can be placed within a structural extracellular matrix (ECM), e.g. a collagen scaffold. They are widely used for studying e.g. drug response, tissue fluidity and tumor invasion [11–13]. On the *in silico* side, tumor growth models of varying degree of coarse-graining are being developed [14–16], some of which are also applied to simulate tumor spheroids [17, 18]. Thus, both experimentalists and theorists generate data for the same systems, but these studies are often not compared quantitatively. Quantitative comparison is an important step towards fully leveraging the results of both groups, and in this context requires an adaptive and robust comparison strategy for spheroid data, regardless of its origin. In prior studies, Browning et al. have investigated structural aspects of melanoma-derived spheroids by building a data-analysis pipeline for spheroid images resulting from confocal microscopy [19], and Szymańska et al. have studied the proliferation behavior of spheroids using Bayesian inference [20]. However, to our knowledge there is currently no strategy for systematically comparing 3D structural data between two spheroids, which can be obtained *in vitro* using stacked multiphoton microscopy images [13]. Hence, in this study, we

want to provide a toolbox of features which may be extracted from a given 3D structure of a spheroid, and metrics to compare these extracted features between different spheroids. These features and metrics can be used on their own, or in combination, to obtain an overall deviation score. Our strategy utilizes point cloud data, in which each point denotes the position of a cell. Importantly, this enables the comparison of both simulated and experimental spheroids. To demonstrate this, we applied our toolbox to previously published data that captured structural differences in triple-negative breast cancer spheroids invading into a collagenous ECM of varying density [13]. Such experimental data were used as a motivation to simulate a variety of spheroid behaviors *in silico* (see Fig 1). Using our previously developed platform “Cells in Silico” (CiS) [21], we performed multiple simulations of 3D tumor spheroids, and for this study, we want to highlight a subset of four spheroid phenotypes: “spherical”, “spherical with far gaslikes”, “deformed”, and “disordered”. These phenotypes emerged from different combinations of the aforementioned parameters (see Sections 2.1 and 4.2), and will be used as examples throughout.

In the following, we will first outline our work with CiS towards simulating spheroids and arriving at the four phenotypes. Then, we will describe the spatial features that we extracted from individual simulated and experimental spheroids. Next, we will discuss our strategy for comparing these features between multiple spheroids, including the derivation of an overall deviation score, and how it can be tuned for a specific use case. After validating our strategy via a transformation study, we will show comparisons between exclusively simulated spheroids, exclusively experimental spheroids, and comparisons between simulated and experimental spheroids. We will conclude by evaluating the success of our method, and providing an outlook for its further use.

## 2 Results

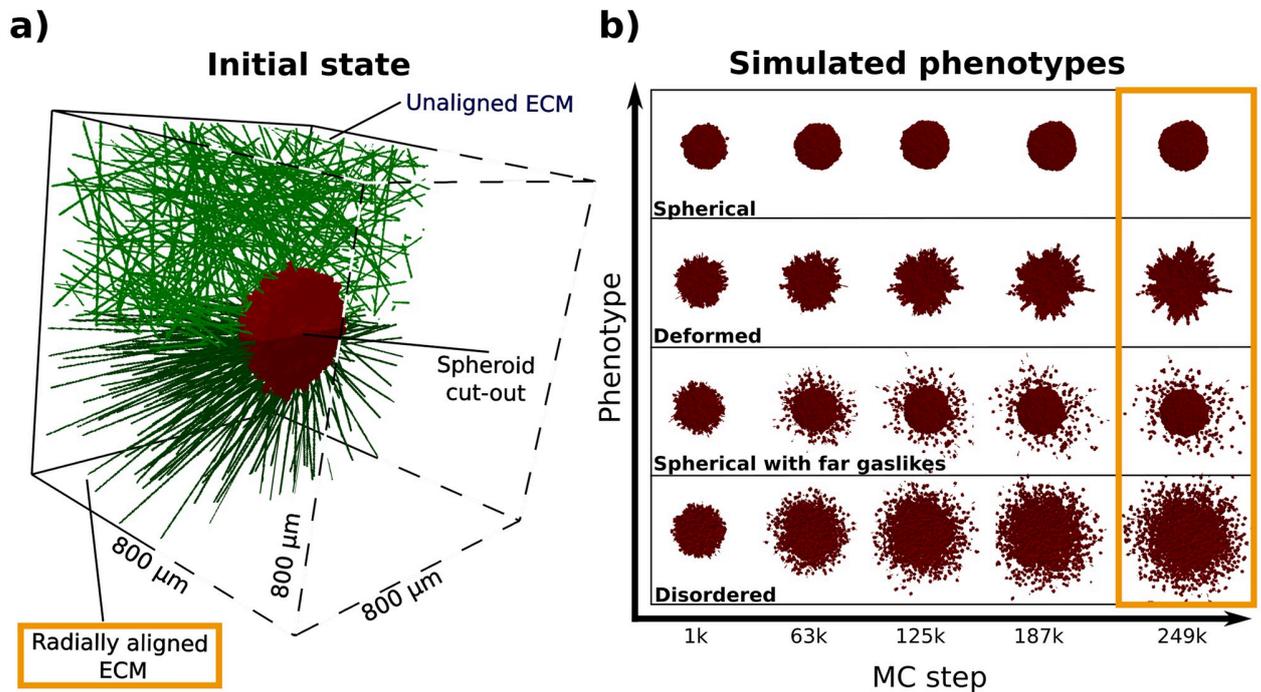
### 2.1 Adapting CiS to the simulation of spheroids

CiS is a highly scalable general-purpose framework for tissue simulation at subcellular resolution. It extends a Cellular Potts Model (CPM) with an agent-based layer, and allows the description of various properties and phenomena, such as cell-cell adhesion, cell compressibility and cell motility, cell division, cell mutation and cell-ECM interactions (see Section 4.1 for more details). In order to apply it to the simulation of spheroids, we first defined the simulation system and parameters to be investigated. Previous studies have identified cell-ECM interactions such as adhesion, degradation and remodeling as strong components in facilitating invasion [22–24]. Hence, we focused on the effects of different ECM alignments, cell-ECM adhesion, ECM degradation, as well as self-propelled cell motility (see Section 4.2 for more details). We performed multiple simulations of 3D tumor spheroids placed in a coarse-grained, rigid ECM. In our model, this ECM can be both adhered to and degraded by cells, but due to its rigidity, the alignment of the fibers remains constant (see Section 4.2 for more detail). Since it is known that tumors remodel their ECM, and ECM alignment is one of the main drivers of invasion [24], we decided to include a radially aligned ECM in our studies (see Fig 1a). Thus, we performed multiple simulations of spheroids at different model parameters (see Table 2). Within this parameter space, we observed four spheroid phenotypes, which were connected by singular differences in parameter values (see Table 1). These were the following:

**Spherical.** Cell-cell adhesion dominated, and the cells remained in a spherical arrangement, with a relatively smooth surface of the spheroid bulk throughout the simulation.

**Deformed.** Due to a strong increase of the cell-ECM adhesion strength, cells adhered to and moved along the radially aligned fibers. Through a combination of cell division filling

## Spheroid simulations using Cells in Silico



**Fig 1. Simulated spheroids and emerging phenotypes.** a) Cut-outs of the initial states of simulated spheroids for two ECM alignments. Each spheroid, shown in red and containing roughly 2000 cells, was placed into an  $(800 \mu\text{m})^3$  volume and surrounded by either an unaligned or radially aligned ECM (green fibers). To improve visibility, the front half of the volume (dashed lines) is not shown b) Time evolution of simulated spheroids displaying four different phenotypes: “spherical”, “deformed”, “spherical with far gasliques”, “disordered”. The ECM is radially aligned for these phenotypes, and is not shown in order to highlight the spheroid morphology. Each phenotype resulted from different combinations of parameters connected to the cell motility, the cell-cell adhesion and the interaction with the ECM (see Section 2.1 and Table 1). Each simulation lasted 250 000 Monte-Carlo (MC) steps, and shown are five snapshots for each simulation. A single MC step corresponds to roughly 1 s of real time in the context of this study. Throughout our investigation, we focused on the final configuration (orange rectangle), and used five replicates from each phenotype.

<https://doi.org/10.1371/journal.pcbi.1010471.g001>

gaps and cell-cell adhesion keeping the bulk intact, the spheroid lost its spherical shape and exhibited protrusions along the ECM fibers.

**Spherical with far gasliques.** Through a combination of high ECM degradation rate and cell-ECM adhesion, cells in the outermost layer of the spheroid moved into the ECM faster than by adhesion alone. This resulted in a halo of singular cells around an approximately spherical spheroid bulk.

**Table 1. Differences between the four simulated phenotypes used throughout this study.**

Phenotype	Cell-ECM adhesion	ECM degradation period	Motility magnitude
Spherical	50	$\infty$ (disabled)	0
Deformed	450	$\infty$ (disabled)	0
Spherical with far gasliques	450	5000 MC steps	0
Disordered	450	5000 MC steps	100

Listed are the three parameters in which the phenotypes differed from each other. The remaining parameters were the same for all phenotypes, and were as follows: ECM density: 1; ECM alignment: radially aligned; Cell-cell adhesion: 50; Random walk persistence: 0; cell division: enabled.

<https://doi.org/10.1371/journal.pcbi.1010471.t001>

**Table 2. Simulated parameter space from which the four phenotypes were obtained.**

Parameter	Values
ECM density	1, 2
ECM degradation period	1000 MC steps, 5000 MC steps, $\infty$ (disabled)
ECM alignment	unaligned, radially aligned
Cell-cell adhesion	50, 100
Cell-ECM adhesion	50, 450
Motility magnitude	0, 50, 100
Random walk persistence	0, 0.3, 0.5
Cell division	enabled

Listed are the number of values per parameter, as well as the total number of parameter combinations resulting from this (see section 4.2 for more detail). For each simulation, five replicates were simulated.

<https://doi.org/10.1371/journal.pcbi.1010471.t002>

**Disordered.** The cells dissociated from each other due to high self-propelled motility. Subsequently, the spheroid integrity was lost.

These four phenotypes visually differed between each other (see Fig 1, and the next step of our investigation was to quantify these differences.

## 2.2 Individual features

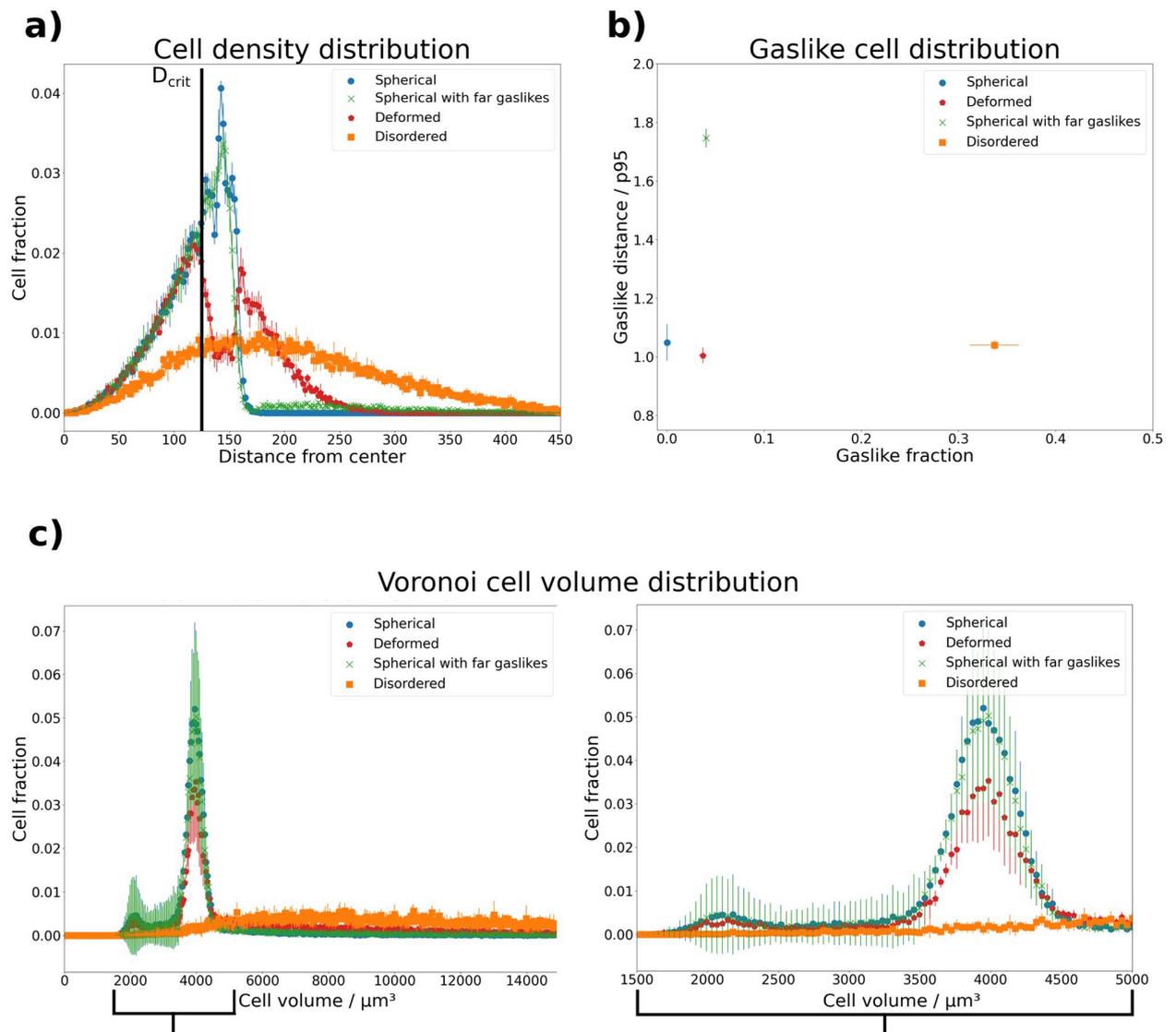
We focused on the analysis of spatial properties of tumor spheroids. For this, we utilized data containing the three-dimensional positions of all individual cell centers at one given point in time. For the analysis of these data we considered various features which could be used individually or as an overall deviation score (see Section 2.4). These features and their applicability are highlighted in the following.

**Cell density distribution.** Analyzing the distribution of cell density is useful for determining the extent of the bulk of the spheroid, as well as its geometry. For our studies we focused on the so-called central local density, which we defined as the fraction of cells found within spherical layers of constant thickness and increasing radii around the spheroid center. For a uniform spherical distribution of cells, this density is non-zero only within the spheroid bulk (see Fig 2a). Disordered spheroids, on the other hand, exhibit a distribution over a larger domain.

**Gaslike cell distribution.** The detachment of single “gaslike” cells from a spheroid has been a recent focus [13], and can be used to distinguish between ordered and disordered spheroids. However, the assignment strategy of the “gaslike” status needs to be well defined. Kang et al. were able to experimentally measure the spheroid boundary [13], and defined cells outside of this boundary as “gaslike”. Since the spheroid boundary is not tracked in our simulations, we used a definition based on nearest-neighbor distances and distance from the center of the spheroid. The set of gaslike cells  $G$  as a subset of all cells  $C$  is thus defined as follows:

$$G = \{c_i \in C : d(c_i, O) > D_{\text{crit}} \wedge \min(N_i) > d_{\text{crit}}\}, \quad (1)$$

where  $D_{\text{crit}}$  is the threshold distance from the spheroid center  $O$ ,  $N_i = \{d(c_i, c_j) | c_j \in C, c_j \neq c_i\}$  is the set of Euclidean distances between cell  $c_i$  and all other cells, and  $d_{\text{crit}}$  is the threshold neighbor distance. The first constraint in Eq 1 provides the context of a bulk structure, and its parameter  $D_{\text{crit}}$  can be selected considering the inflection point of the central local density. The second constraint ensures that only detached cells are defined as gaslikes, and its parameter  $d_{\text{crit}}$  was chosen by considering the mean distance between all cells. For our purposes we



**Fig 2. Visualization of cell based features extracted from simulation data for four different phenotypes.** a) **Cell density distribution.** Shown is, averaged over all replicates of each phenotype, the fraction of cells within spherical layers around the spheroid center *versus* the radii of these layers. The “spherical” phenotype shows a steep drop at a radius of 150  $\mu\text{m}$ , while the “deformed” and “disordered” phenotypes show a long-tailed distribution. The “spherical with far gasliques” phenotype behaves similar to the “spherical” phenotype, except for a non-zero density above 175  $\mu\text{m}$ . b) **Gaslike cell distribution.** Shown are the average fractions of gaslike cells according to Eq 1 *versus* their normalized average distance to the spheroid center. The fraction of gasliques exhibited by the “spherical”, “spherical with far gasliques” and “deformed” phenotypes is similar, but the distance from the spheroid center is far greater for the “spherical with far gasliques” phenotype. The “disordered” phenotype on the other hand contains many cells classified as gasliques across the entire spheroid volume. Their normalized average distance from the center evens out to a value slightly above 1. c) **Voronoi cell volume distribution.** Shown are histograms of the average Voronoi cell volumes found in the four phenotypes. The “spherical” phenotype shows a sharp peak around a volume of 4000  $\mu\text{m}^3$ , and a smaller peak around a volume of 2000  $\mu\text{m}^3$ . The “spherical with far gasliques” and “deformed” phenotypes show a similar behavior, with a slightly more pronounced tail towards larger volumes. Finally, the “disordered” phenotype shows volumes distributed over a wide range. The range between volumes of 1500  $\mu\text{m}^3$  and 5000  $\mu\text{m}^3$  is magnified on the right to highlight the differences between phenotypes in the two peaks.

<https://doi.org/10.1371/journal.pcbi.1010471.g002>

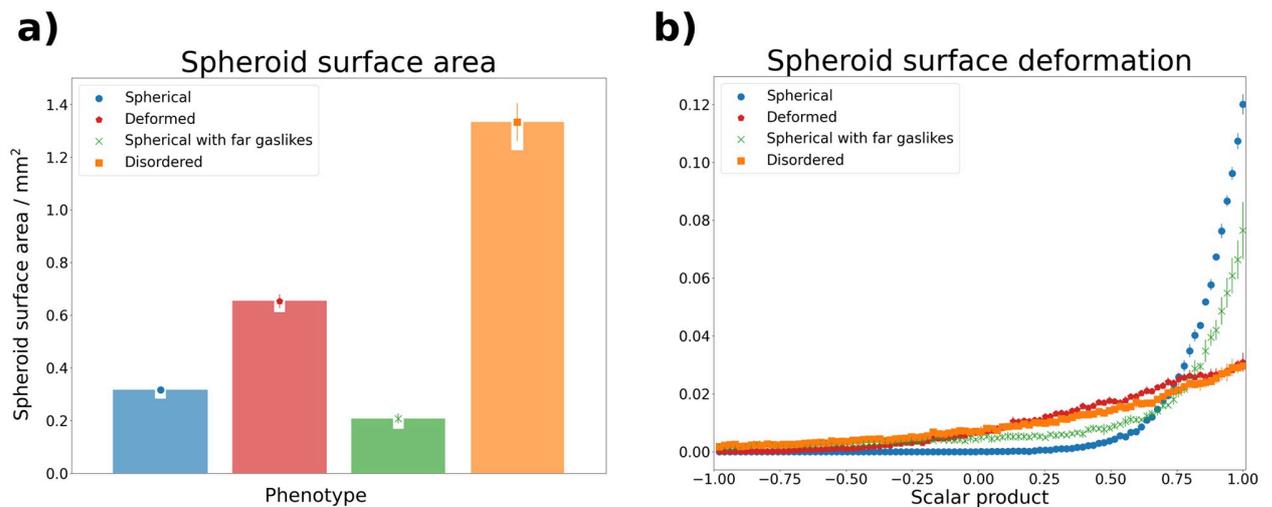
selected the data by Kang et al. [13] as basis for deriving these parameters:  $D_{\text{crit}} = 125 \mu\text{m}$ ,  $d_{\text{crit}} = 19 \mu\text{m}$ . This set  $G$  can be used to compute various properties, such as the fraction of gaslikes and their average distance from the spheroid center. We combined these two properties in this feature, defining it as a point  $p$  within the space spanned by them. The first property  $p_x$  describes the fraction of all cells in the spheroid that are detached. The second property  $p_y$  describes the mean distance of the detached cells from the spheroid center.

$$p = (p_x, p_y) = \left( \frac{|G|}{|C|}, \frac{1}{|G|} \sum_i \frac{d(c_i, O)}{p_{95}(G^*)} \right) \quad (2)$$

where  $p_{95}(G^*)$  is the 95th percentile of distances of the non-gaslike cells  $G^* = C \setminus G$  from the spheroid center, which serves as a normalization factor. We included only the non-gaslike cells for this normalization factor, because our aim was to define the distance relative to the spheroid bulk. As shown in Fig 2b, the “disordered” and “spherical with far gaslikes” phenotypes can be distinguished from the others using this feature, but it is suited less well for comparing “spherical” and “deformed” spheroids.

**Voronoi cell volume distribution.** The distribution of Voronoi cell volumes within the spheroid serves as a measure of cell deformation, as well as their confinement. To obtain these volumes, we performed a Voronoi tessellation [25] on the cell center point cloud, during which the system was divided into  $N_{\text{cells}}$  regions according to the distances between adjacent cells. It is important to note that the Voronoi cell volumes are not the same as the biological cell volumes, but represent a proxy in which detached cells occupy a significantly larger volume. We generated a histogram of the Voronoi cell volumes, as shown in Fig 2c). Here we observe that the first three phenotypes are distributed sharply around a volume of roughly  $4000 \mu\text{m}^3$ , with a smaller spike around  $2000 \mu\text{m}^3$  and a tail. The volumes of the “disordered” phenotype are evenly distributed over a much wider range. The tail of the first three distributions is a useful artefact of the Voronoi tessellation, as it allows to extract additional information about the bulk spheroid surface.

**Spheroid surface and surface deformation.** While the cell density distribution provides a measure of the spheroid size, its information about the spheroid shape is limited. To study this in more detail, we needed to approximate the spheroid surface, as we wanted to distinguish deformed spheroid bulk from spherical bulk. We did this via surface triangulation using the marching cubes algorithm [26]. To apply this algorithm we performed some preprocessing of the point cloud data: first, we extracted the set of non-gaslikes  $G^*$ , as we were only interested in the shape of the spheroid bulk. Next, we obtained a continuous spheroid volume from the remaining points by voxelizing our data. This voxelization was performed by mapping the system onto a 3-dimensional density grid. The density of each grid point was set to 1 if the grid point was closer than a threshold distance to any cell in the point cloud, and was set to 0 otherwise. The resulting region defined by grid points of a density of 1 served as a proxy of the continuous spheroid volume. We then used the marching-cubes algorithm [26] to generate a mesh of triangles which approximated the surface of the continuous volume, a process known as surface triangulation (see also S4 Fig). Finally, we extracted two features from this: first, we calculated the surface area from the triangle mesh (see Fig 3a)), and second, we analyzed the surface deformation by investigating the orientations of the mesh vertices. This was done by calculating, for each vertex, the scalar product between its normal vector and its origin vector, with the spheroid center at the origin (see S4 Fig). Then these scalar products were combined in a histogram. The vertex orientations serve as a measure of deformation, since for a perfect sphere all scalar products are equal to 1, and a deformed sphere results in a more widely spread distribution (see Fig 3b).



**Fig 3. Visualization of spheroid bulk based features extracted from simulation data for four different phenotypes.** Surface information was extracted via the marching cubes algorithm [26] (see also section 2.2). **a) Spheroid surface area.** Shown is the average surface area found for each phenotype. The “spherical with far gasliques” phenotype has the smallest average surface area, due to the spherical bulk containing less cells than that of the “spherical” phenotype. The larger average surfaces of the “deformed” and “disordered” phenotypes are due to their more irregular shape. **b) Spheroid surface deformation.** Shown are histograms of the scalar products between vertex normal vectors and vertex origin vectors, with the origin denoting the center of the spheroid. The vertices were obtained from surface triangulation of the spheroid point cloud and denote points on this surface (see also S4 Fig). The “spherical” phenotype exhibits a sharp peak at scalar products of 1, which is less pronounced for the “spherical with far gasliques” phenotype. The remaining two phenotypes are spread more widely.

<https://doi.org/10.1371/journal.pcbi.1010471.g003>

By using these features we were able to measure and quantitatively describe different aspects of individual tumor spheroids. This provided a basis on which we could compare two spheroids with each other. Such a comparison required the definition of distance metrics for each feature, which are highlighted in the following.

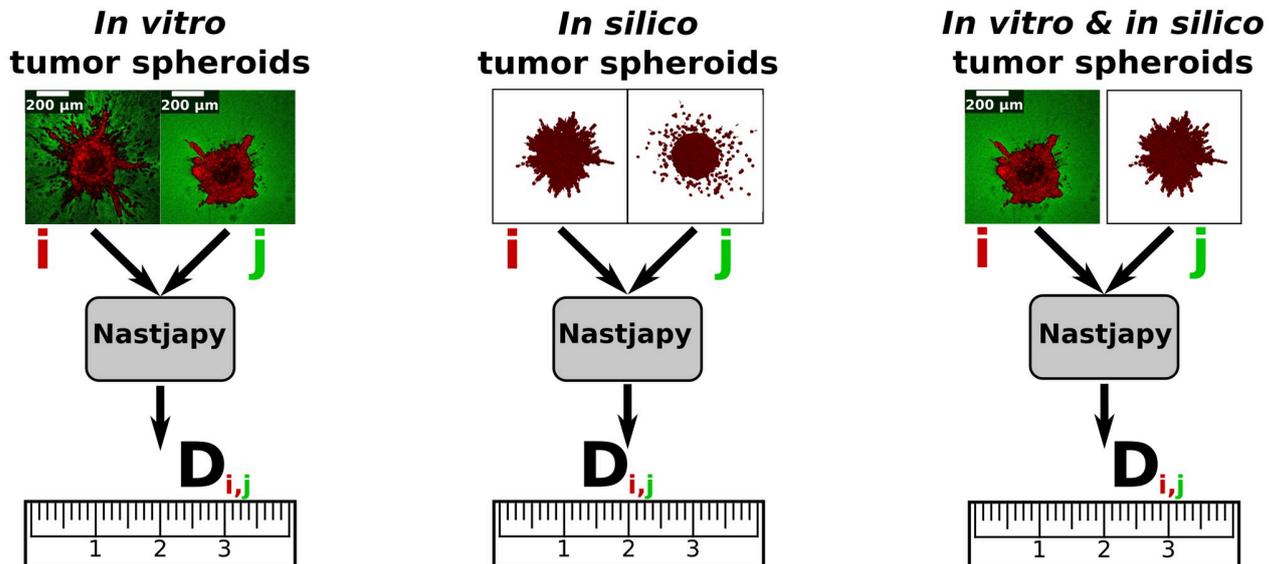
### 2.3 Individual metrics

To accommodate the different types of output data between features, we required suitable metrics. For the spheroid surface area, which provided a scalar value per spheroid, we used the mean squared error (MSE). For the gaslike cell distribution, which provided a tuple of two coordinates per spheroid, we used the Euclidean distance. Finally, for the distribution-based features, i.e. cell density distribution, Voronoi cell volume distribution and spheroid surface deformation, we used the 1-Wasserstein distance (WSD). The Wasserstein distance is a metric between probability distributions, and is a common sight in mathematics, especially statistics and computer science. The  $p$ -Wasserstein distance between two probability measures  $\mu$  and  $\nu$  on the metric space  $(\mathbb{R}^n, d)$  is defined as:

$$W_p(\mu, \nu) := \left( \inf_{\gamma \in \Gamma(\mu, \nu)} \int_{\mathbb{R}^n \times \mathbb{R}^n} d(x, y)^p d\gamma(x, y) \right)^{1/p} \quad (3)$$

where  $\Gamma$  denotes the collection of all joint probability measures  $\gamma$  with marginals  $\mu$  and  $\nu$  [27], and  $d(x, y)$  denotes the metric distance used to define the metric space. Here,  $d(x, y) = |x - y|$ . An intuitive illustration of the Wasserstein distance can be given by viewing each distribution as a pile of earth of different shape, and considering the amount of work that has to be done to transform one pile into the other. Assuming this work to be equal to the product between the amount of earth that has to be moved and the distance it needs to be moved, the

## Nastjapy as a spheroid comparison tool



**Fig 4. Sketch of the application of *Nastjapy*.** An overall deviation score  $D_{i,j}$  is calculated between two individual spheroids  $i$  and  $j$  (see also section 4.4).

<https://doi.org/10.1371/journal.pcbi.1010471.g004>

Wasserstein distance is the minimum amount of work that has to be done. Due to this illustration, the WSD is often referred to as “earth mover’s distance” [28].

### 2.4 Combination of multiple metrics

At this point, the individual features described in the previous sections could be reliably compared between two spheroids using our defined metrics. Next, one of our main goals was to combine these features and metrics into a single scalar value, which could then serve as an overall deviation score between two spheroids (see Fig 4). This can be seen as a summary statistic, which are also used in Bayesian inference [29]. Many different questions regarding the comparison of tumor spheroids require such a singular scalar. From an experimentalist’s view, this could be the comparison of spheroids cultivated in different conditions, with the goal of quantitatively determining how the change of one experimental variable influences the spheroid growth and invasion pattern. A problem faced by theorists running simulations is how to optimize the model parameters to reproduce experimental results. Both problems require one scalar distance measure like the one we aimed to derive here. Before doing so, we need to address the fact that it is unlikely for such a distance measure to be generally applicable to all types of tumor spheroids and experimental settings. This is due to the high dimensionality of even a single spheroid dataset at a single point in time. To illustrate this, we consider the case of comparing two spheroids, each containing 1000 cells. The desired distance is a function  $f : \mathbb{R}^{3000} \times \mathbb{R}^{3000} \rightarrow \mathbb{R}$ . Such a function will, by design, project many different pairs of spheroids onto the same point in  $\mathbb{R}$ . This property can hardly be circumvented, and is desired in a distance measure. On the other hand, this also means that the measure has to be carefully selected depending on the use case. Therefore, in addition to defining the overall deviation score here, we will also propose a method to adapt the score to different use cases.

**Standardization.** We first ensured that all metrics were on a similar scale. For this standardization, we used five replicates from each of the four phenotypes from our simulations, and compared each feature, resulting in  $N_{\text{spheroids}} = 20$  spheroids and  $N_{\text{distances}} = 400$  metric distances per feature  $f$ . These distances  $d_{i,j,f}$  between spheroid  $i$  and spheroid  $j$  were then transformed according to Eq 4:

$$d_{i,j,f,\text{std}} = \frac{d_{i,j,f} - \mu_{d,f}}{\sigma_{d,f}} \tag{4}$$

where  $\mu_{d,f}$  and  $\sigma_{d,f}$  respectively denote the mean and standard deviation across the  $N_{\text{distances}}$  values for each feature  $f$ . Since this standardization may lead to values of  $d_{i,j,f,\text{std}}$  below zero, and we aimed to define a positive distance for each feature, we further shifted each value by the minimum across all  $d_{i,j,f,\text{std}}$ , finally arriving at  $d_{i,j,f^*}$  as defined in Eq 5:

$$d_{i,j,f}^* = d_{i,j,f,\text{std}} + |\min([d_{1,1,f,\text{std}}, d_{1,2,f,\text{std}}, (\dots), d_{N_{\text{spheroids}},N_{\text{spheroids},f,\text{std}}])| \tag{5}$$

**Overall deviation score and use case adaptation.** Next, we defined the overall deviation score  $D_{i,j}$ . This definition entailed merging the previously standardized  $d_{i,j,f}^*$  via the following linear combination:

$$D_{i,j} = \sum_{f=1}^{N_{\text{features}}} \lambda_f \cdot d_{i,j,f}^*, \tag{6}$$

where  $\lambda_f$  denote the weight factors for each feature, i.e. how much it contributes to the final deviation score  $D_{i,j}$ . In order to optimize these values, we once again turned to our simulated phenotypes and their five respective replicates. Because we wanted to ensure the deviation score to distinguish between different phenotypes, we required values of  $\lambda_f$  that minimized the intra-phenotype deviations and maximized the inter-phenotype deviations. This can be formulated as a maximization problem:

$$\max_{\{\lambda_f\}} \sum_{f=1}^{N_{\text{features}}} \lambda_f \sum_{k=1}^{N_{\text{phenotypes}}} \left( \underbrace{\sum_{i \in P_k} \sum_{j \notin P_k} d_{i,j,f}^*}_{\text{inter-phenotype}} - \underbrace{\sum_{i \in P_k} \sum_{j \in P_k} d_{i,j,f}^*}_{\text{intra-phenotype}} \right), \tag{7}$$

where  $P_k$  is the set of all spheroids of phenotype  $k$ . This optimization procedure can be interpreted as an inverse clustering. During clustering, the property described in Eq 7 is maximized by assigning individuals to a cluster. In contrast to this, our method uses prior clustering information to optimize the metric space itself. This shows resemblance to methods of contrastive learning, with the notable difference that we use a linear model in our approach [30]. We argue, that adjusting the weighting of features according to their relevance to the formation of predefined clusters will allow to more strongly distinguish between those clusters. Assuming that phenotypes are correctly grouped, maximizing Eq 7 already ensures that each  $\lambda_f > 0$ . Additionally, we decided on the following constraint:

$$\sum_{f=1}^{N_{\text{features}}} \lambda_f^2 = 1 \tag{8}$$

This constraint is important to prevent the optimization procedure from collapsing towards the trivial solution of setting  $\lambda_f \rightarrow \infty$ . It also fixes each  $\lambda_f$  to the domain  $[0, 1]$ . Furthermore,

**Table 3. Fitted weight factors for each feature contributing to the overall deviation score between two spheroids.**

Feature	Cell density distribution	Gaslike cell distribution	Voronoi cell volume distribution	Spheroid surface area	Spheroid surface deformation
$\lambda_f$	0.41	0.50	0.43	0.34	0.52

The values were obtained by maximizing Eq 7, with the simulated phenotypes serving as a calibration set (see Section 2.1).

<https://doi.org/10.1371/journal.pcbi.1010471.t003>

using the square of each  $\lambda_f$  minimizes the influence of outliers across features. The optimization of Eq 7 under consideration of this constraint was performed using the Sequential Least Squares Programming (SLSQP) method implemented in the SciPy package [31]. The contribution of each feature to the overall deviation score resulting from the optimization of the  $\lambda_f$  is shown in Table 3.

## 2.5 Validation: Transformation study

Validating a metric, such as the one derived in this work, requires a set of data examples with known relation. For this purpose, we designed a transformation study, in which we generated multiple point clouds from a reference spheroid using transformation functions. These functions were selected in such a way, that for a useful metric we expected higher distances between reference and transformed point cloud for higher transformation strengths. On the other hand, the metric has to remain invariant under transformations related to the frame of reference, e.g. rotation or translation of the point cloud. Therefore, we also included these transformations. Hence, we tested for the following properties:

1. invariance under rotation and translation,
2. monotony within the domain of interest: a small deviation from an original spheroid shall result in a lower distance than a large deviation.

We investigated these properties for the five described features and the overall deviation score by applying four transformations to a spheroid  $S = \{\vec{P}\} \mid \vec{P} \in \mathbb{R}^3$  of the “spherical” phenotype. The transformations were represented by functions in the space of point clouds  $T : S \times \mathbb{R} \rightarrow S$ . With this approach, we aimed to verify both of the above properties. Invariance is shown, when the distance does not depend on the strength of the transformation. Similarly, monotony is shown, when the distance metric grows monotonously with the transformation strength. The four transformations that we used were the following:

**Rotation.** Rotating each cell of the spheroid by a given angle  $\alpha$  around an arbitrary axis  $i$  through the spheroid center:

$$T_R(S, \alpha) = \{R_i(\alpha)\vec{P} \mid \vec{P} \in S\}, \quad (9)$$

where  $R_i(\alpha)$  is the rotational matrix.

**Noise.** Adding a random vector drawn from a standard uniform distribution to the position of each cell:

$$T_N(S, \alpha) = \{\vec{P} + \alpha \cdot \vec{X} \mid \vec{P} \in S, \vec{X} \sim \mathcal{U}_3(0, 1)\}, \quad (10)$$

**Deformation.** Translating each cell along the radial vector of the spheroid, modulated by the spherical angles of the cell’s position:

$$T_{D,\omega}(S, \alpha) = \{\vec{P} + \alpha \cdot \hat{e}_r \cdot (\cos \omega\phi + \sin \omega\theta) \mid \vec{P} \in S\} \quad (11)$$

This deformation can be interpreted as adding ripples with frequency  $\omega$  and amplitude  $\alpha$  to the spheroid surface.

**Scaling.** Multiplying the position of each cell by its distance to the center of the spheroid:

$$T_S(S, \alpha) = \{\alpha \cdot \|\vec{P}\| \cdot \vec{P} \mid \vec{P} \in S\} \quad (12)$$

This transformation affects cells with a larger distance to the spheroid center more strongly than those close to it. The spheroid density is therefore not conserved.

A visual example for each of these four transformations is provided in Fig 5. For completeness, we also mention the translation transformation, which is defined as follows:

**Transformation.** Changing the position of each cell by the same vector  $\vec{\alpha}$

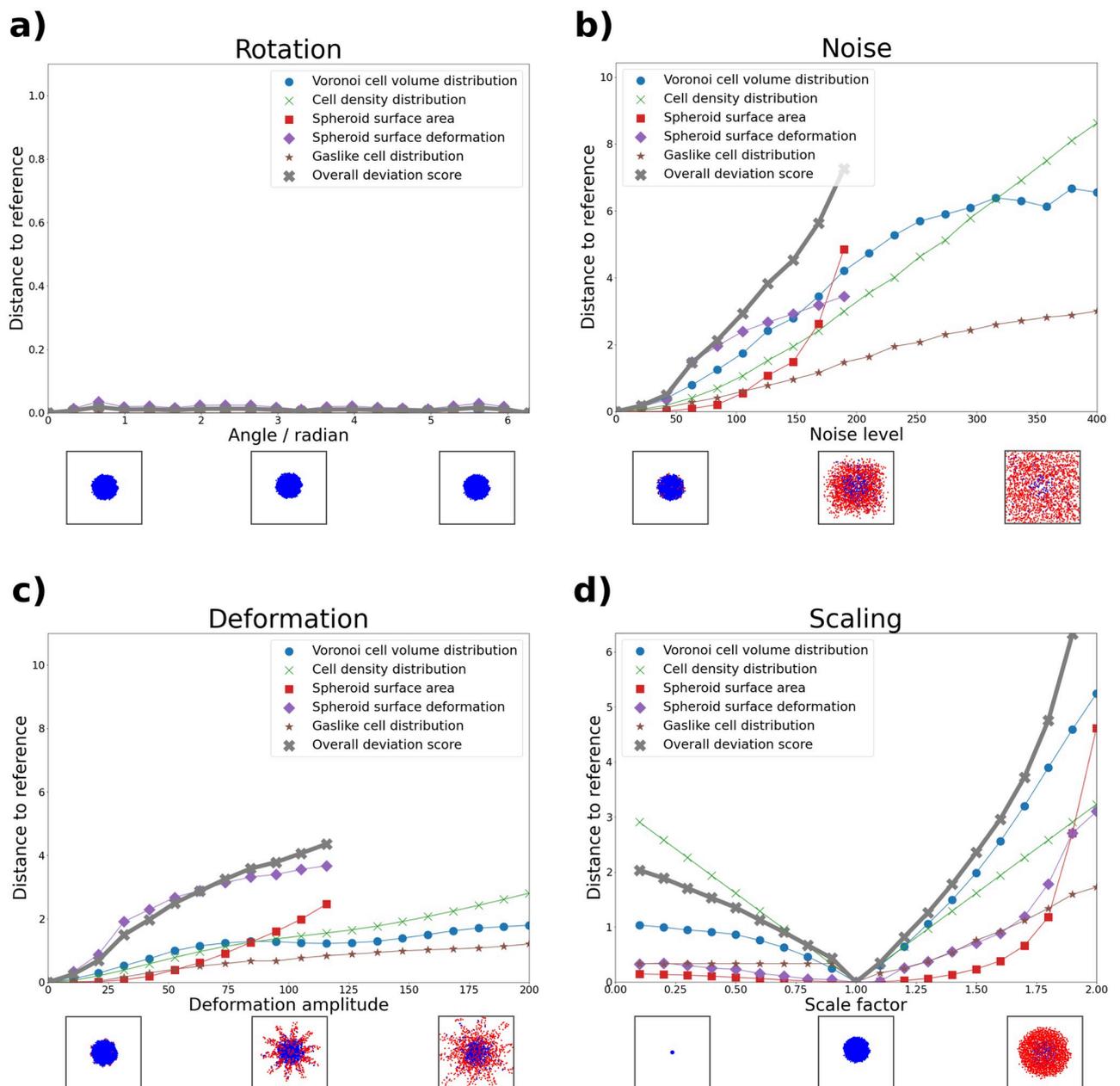
$$T_T(S, \vec{\alpha}) = \{\vec{P} + \vec{\alpha} \mid \vec{P} \in S\} \quad (13)$$

We do not show it however, because translational invariance is ensured. This is because all features depend only on relative and not absolute distances, and this transformation conserves relative distance. Rotational invariance was expected due to the rotational symmetry of the underlying features. Nonetheless, we wanted to test whether artifacts, produced by the voxelization for features related to the spheroid surface (see Section 2.2), had any notable effect. For this reason, we included the rotation transformation. The remaining transformations were chosen to validate the monotony of the deviation score.

We applied each transformation at increasing strength and compared the resulting spheroid with the untransformed version. The results of this are shown in Fig 5. Starting with the rotation transformation in subfigure a), we observed no change in the feature distances at increasing rotation angle, except for negligible changes in the spheroid surface deformation. This underlines the rotational invariance of our features. For the other three transformations, we observed monotony in all cases. Those features related to the spheroid surface could not be meaningfully extracted when the spheroid bulk was disrupted, i.e. at high degrees of the noise and deformation transformations (subfigures b) and c)). The gaslike distribution feature distance remained constant for small values of the scaling transformation (subfigure d)), since no cells were classified as gaslike here. Aside from these edge-cases, our features behaved robustly. It is interesting to note that the overall deviation score scaled approximately linearly with the transformation strength within the domain of interest, excluding the aforementioned extreme cases. This property can be viewed as a stronger version of the monotony property. Importantly, this was not used as a constraint when optimizing the weights, but emerged from the procedure itself.

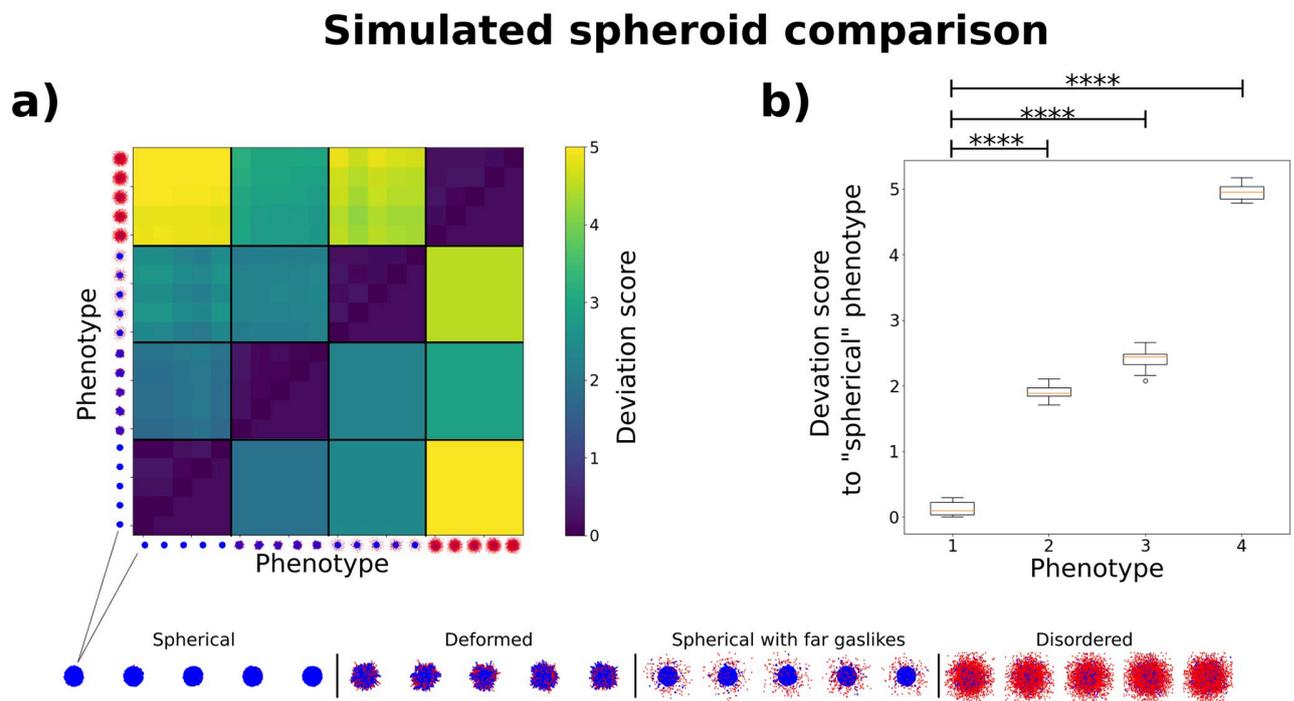
## 2.6 Validation: Comparing simulated spheroid phenotypes

As a second way to validate our methods, we now moved to the comparison of simulated spheroids. We chose the final simulation state, after 250 000 MC steps, of five new replicates from each phenotype. Importantly, these were not the same replicates which we used earlier for the calibration of the weight factors. We calculated the overall deviation score for each pair. As shown in Fig 6a), we compared individual replicates (upper triangle), and we also combined replicate comparisons into an average phenotype deviation score (lower triangle). Importantly, while it is questionable whether the “disordered” phenotype is a biologically occurring configuration (see section 3), we chose to include it to serve as a phenotype maximally distant to the others. The deviation score was lowest when a phenotype was compared with itself, and highest, when any phenotype was compared with the “disordered” one. The “spherical”, “deformed” and “spherical with far gaslikes” phenotypes showed a smaller deviation score between each other, but were nonetheless distinguishable. This is underlined in



**Fig 5. Feature comparison for spheroid point clouds resulting from four different transformation functions.** Shown are the standardized metric distances between the un-transformed reference spheroid and an increasingly transformed version for each data feature. In addition, the combined deviation score is depicted in gray crosses for each transformation (see Section 2.4). Below each subfigure, we provide a top-down view snapshot of the spheroid at three levels of transformation. Blue cells are classified as non-gaslike, and red cells are classified as gaslike. **a) Rotation.** Except for negligible changes in the spheroid surface deformation feature, we observe no change at increasing rotation angle. This supports rotational invariance of our features. **b) Noise.** For each feature, the distance increases at increasing noise level. Due to the loss of a solid core at high noise levels, the spheroid surface area and deformation features are no longer sensible, and were therefore cut. The deviation score increases approximately linearly up to a noise level of 200, at which point the features related to the spheroid surface area were cut. **c) Deformation.** Similar behavior to b) is observed here. Above a deformation amplitude of 120 the spheroid point cloud still contains cells classified as non-gaslike but loses its solid core. Surface area and deformation values were therefore cut above this threshold. The deviation score increases approximately linearly up to a deformation amplitude of 120. **d) Scaling.** We observe increased distances both for scale factors below and above 1. Due to the fixed values of  $D_{crit}$  and  $d_{crit}$  (see Eq 1), the gaslike distribution feature is scale-dependent, and also varies here. For scale factors below 1.0, no gaslikes were found, and therefore the values of this feature remained constant. The deviation score increases approximately linearly both for scale factors smaller and larger than 1.

<https://doi.org/10.1371/journal.pcbi.1010471.g005>



**Fig 6. Deviation score comparison for four simulated spheroid phenotypes.** a) Shown are the deviation scores for five replicates of each phenotype on the upper triangle, and the average deviation score over all replicates of each phenotype on the lower triangle. A top-down view of the spheroid point cloud for each replicate is shown next to the respective row/column. Blue cells are classified as non-gaslike, and red cells are classified as gaslike. For better comparison, an enlarged version of each spheroid was placed at the bottom of the figure. We observe the highest deviation between the “disordered” phenotype and the others, with the maximum deviation between the “spherical” and the “disordered” phenotypes. The “spherical”, “spherical with far gasliques” and “deformed” phenotypes, which are more similar from a visual perspective, show a smaller deviation score using our analysis, but are nonetheless distinguishable. b) Box plots of the deviation score values between the “spherical” phenotype and each other phenotype. The values used here correspond to those used for the lowest row of subfigure a). We observe that the deviation scores for the “spherical” phenotype compared with the other phenotypes consistently lie above the maximum deviation score of the “spherical” phenotype compared with itself. Significance was determined using Welch’s t-test.

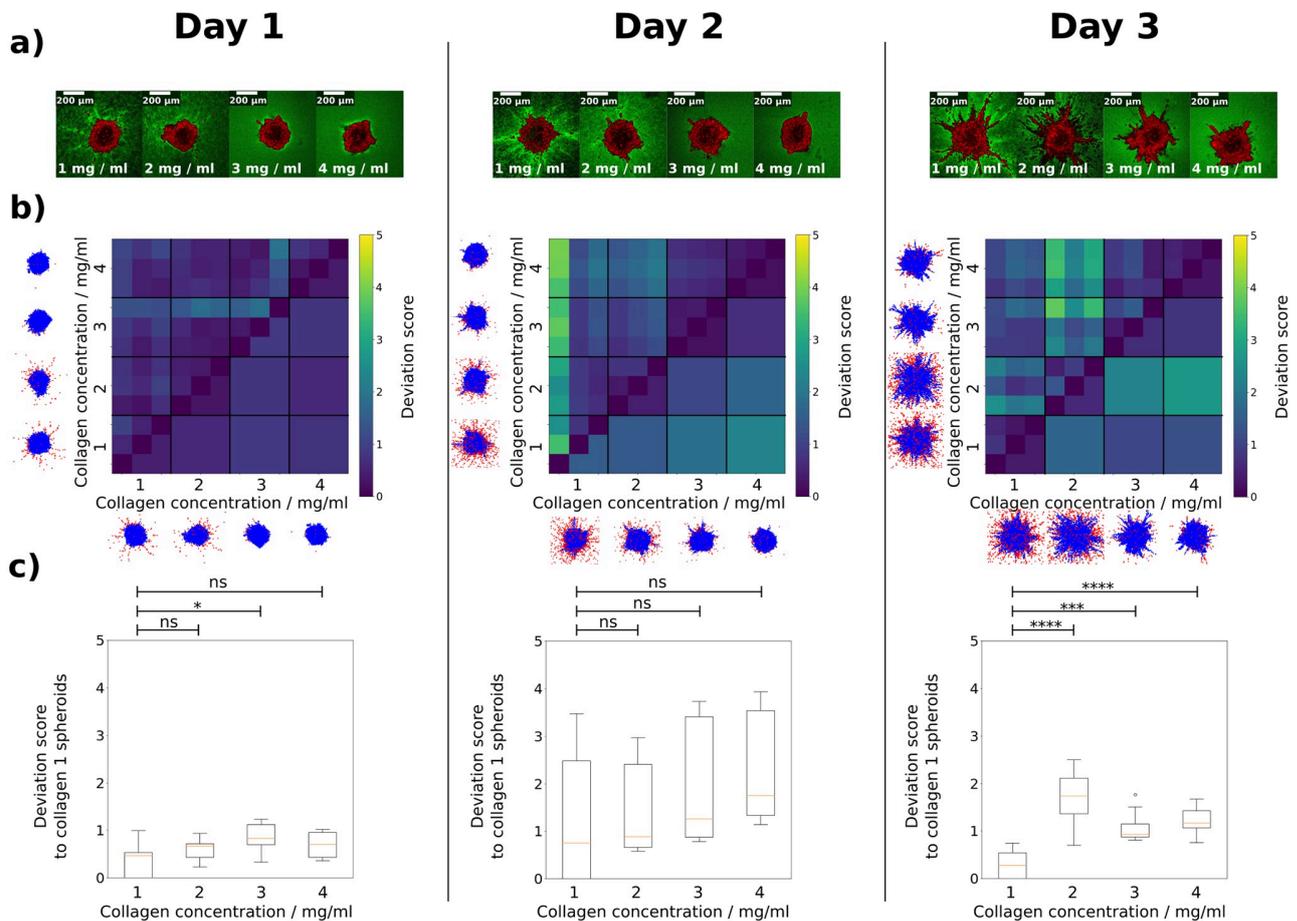
<https://doi.org/10.1371/journal.pcbi.1010471.g006>

subfigure b), in which we show box plots of the overall deviation scores between the “spherical” phenotype and the others. Here, each phenotype comparison was clearly distinct from the comparison of the “spherical” phenotype with itself (see also [S1 Fig](#)). Significance was determined using Welch’s t-test.

### 2.7 Validation: Comparing experimental spheroids

To demonstrate the fact that our deviation score can also be used for experimental data, we again turned to the dataset on which we based our initial simulations. Kang et al. previously investigated the invasive behavior of tumor spheroids cultured in increasing collagen concentrations [13]. Their data set contained cell-resolved 3D snapshots of spheroids in four different collagen concentrations (1–2–3–4 mg/ml) at different times (days 1–2–3) during invasion (see [Fig 7a](#)). For each collagen concentration and day of culture, data from three individual spheroids were acquired using a combination of optical clearing and multiphoton microscopy. Since the optical clearing procedure requires fixation, data from successive days of culture share the same initial conditions but do not originate from the same spheroid. For more details, see Section 4.3.

## Experimental spheroid comparison



**Fig 7. Deviation score comparison for *in vitro* MDA-MB-231 spheroids cultured in four collagen concentrations  $c$  (data provided by Kang et al [13]).** **a)** 2D cross-sections of 3D multiphoton microscopy image stacks depicting one replicate of each collagen concentration [13]. Spheroids were imaged at one, two or three days after embedding in collagen, and were then fixed and imaged (see Section 4.3). **b)** Deviation score comparison between all spheroid samples. For each day, the deviation scores for three replicates of each collagen concentration are shown on the upper triangle, and the average deviation score for each collagen concentration is shown on the lower triangle. A top-down view of the spheroid point cloud for a representative replicate of each collagen concentration is shown next to each heatmap (see S2 Fig for all replicates). Blue cells are classified as non-gaslike, and red cells are classified as gaslike. Due to matching initial conditions, we observe low deviation scores between spheroids grown for one day. These differences increase at day 2, where we observe an approximately linear increase of the average deviation score from  $c = 1$  mg/ml to  $c = 4$  mg/ml. Finally, at day 3, we observe the lowest deviation between  $c = 3$  mg/ml and  $c = 4$  mg/ml. This underlines the findings by Kang et al., who observed a transition in invasion behavior between 2 mg/ml and 3 mg/ml. **c)** Deviation score box plots from spheroids grown for one, two and three days respectively. The box plots for each day show the deviation score values between  $c = 1$  mg/ml and each other concentration. These values correspond to those used for the lowest rows in b). We observe that for spheroids grown for one and two days, the deviation score values of  $c = 1$  mg/ml compared with itself are similar to the deviation score values of  $c = 1$  mg/ml compared with the other concentrations. The differences are nonsignificant and therefore not sufficient to clearly distinguish between the concentrations. This changes at day three, where each concentration shows a higher deviation score to  $c = 1$  mg/ml than  $c = 1$  mg/ml compared with itself. Significance was determined using Welch's t-test.

<https://doi.org/10.1371/journal.pcbi.1010471.g007>

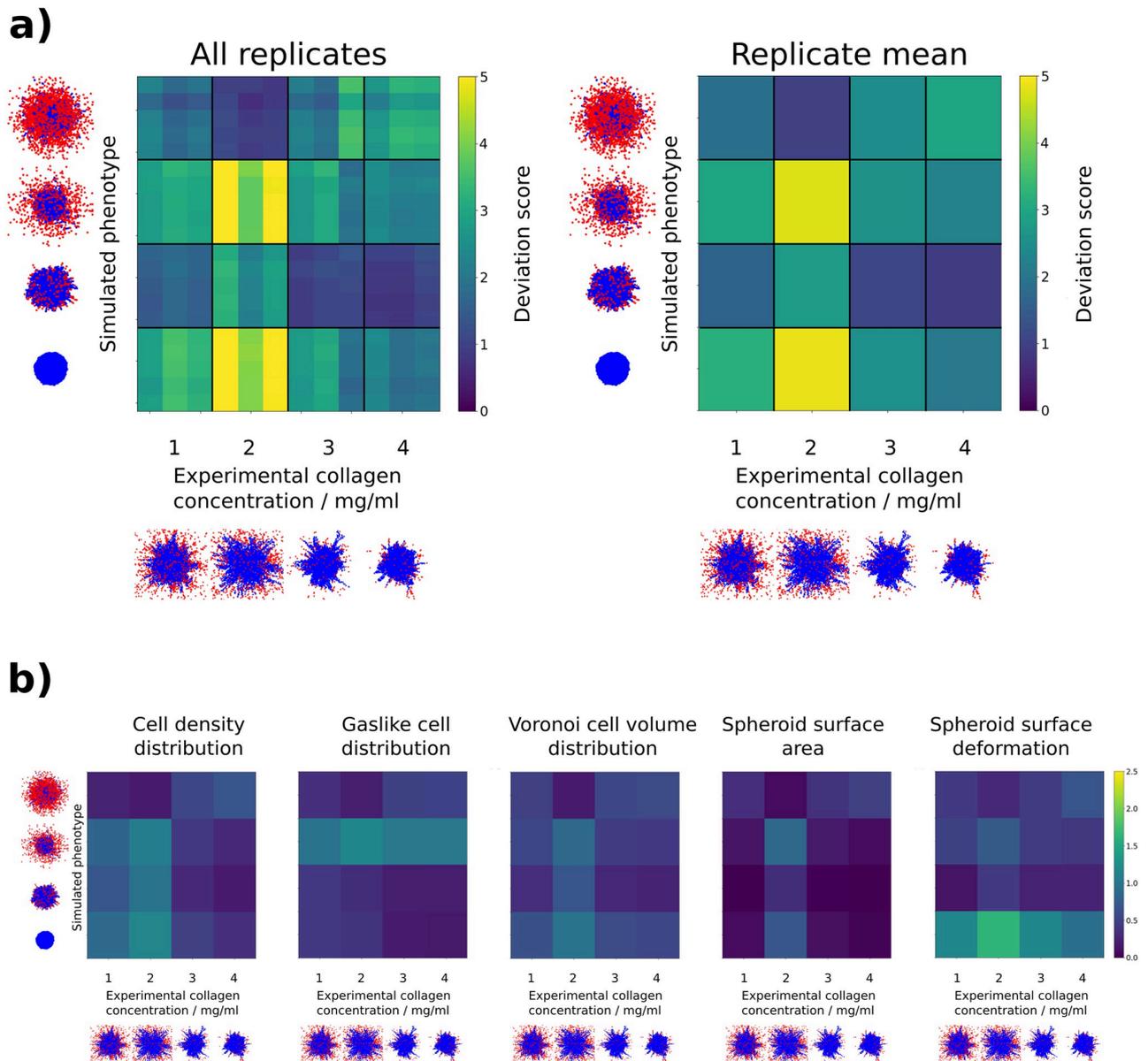
In order to keep the scale consistent throughout this study, we used the same standardization and weight factors as for our simulated spheroids. We compared the spheroids grown for one day, two days, and three days, and visualized them in Fig 7. Similar to Fig 6a), in Fig 7b) we show, for each growth duration, the deviation scores for individual replicates on the upper triangle, and the average deviation score over a set of replicates on the lower triangle. For illustrative purposes and consistency, we have kept the color bar range the same as before. In

subfigure c), we show box plots of the deviation score values between collagen 1 and each other concentration. As expected, we observe low to nonsignificant deviation between the day 1 spheroids, both in subfigure b) and c). This changes at day 2, where the difference between collagen concentrations becomes clearer. However, while the difference is visible in the average deviation score in subfigure b), there is still strong overlap between the values across concentrations. Furthermore, in subfigure c) we see that the differences are not significant. Finally, on day 3, we see the highest difference between the collagen concentration of 2 mg/ml versus the other concentrations, while spheroids of collagen concentrations 3 and 4 mg/ml are most similar to each other. This is consistent with qualitative observations from subfigure a), and from prior quantifications by Kang et al. (cf., Fig 4 c) in [13]) who observed a sudden transition in single cell individualization during invasion between collagen concentrations of 2 and 3 mg/ml. Importantly, as seen in subfigure c), the deviation score reliably distinguishes between the concentrations at day 3.

## 2.8 Comparing simulated and experimental spheroids

Finally, to show another aspect for which the deviation score may be used, we applied our analysis method to the comparison between simulated and experimental spheroids. For this, we used both our simulated phenotypes, and the experimental data from spheroids grown for three days (see Section 2.7). Since the simulation parameters used here were not fitted to the data, but represented default parameter sets, we did not expect a high degree of similarity. On the other hand, this provided an opportunity to investigate both the overall deviation score and the underlying feature distances, and to demonstrate how the differences in spheroid morphology manifested themselves within the features. In subfigure a) of Fig 8, we show the comparison between each experimental replicate (horizontal) and each simulated replicate (vertical) on the left side. On the right side we show the average within replicates. Here, we observed the highest deviation scores between collagen density 2 and both “spherical” (S) and “spherical with far gaslikes” (SFG) spheroids. Visually, this is sensible when comparing the 2D images of the replicates (see also S2 Fig for this); the round shapes of the S and SFG spheroids differ strongly from that of spheroids in 2 mg/ml collagen, as does the number and location of cells classified as gaslike. Furthermore, we observed the lowest deviation scores between spheroids in 4 mg/ml collagen and “deformed” spheroids. Here, the spheroid shape visually matched much better between the replicate images. The deviation score between spheroids in 2 mg/ml collagen and the “disordered” (D) phenotype is also low. These visual differences and similarities are reflected in our features, as seen within subfigure b). Here we decomposed the overall deviation score back into its components, and thereby show the influence of each feature on it. To improve visibility, the color bar range maximum is set to half of that shown in subfigure a). We see that the distance between the collagen density 2 spheroids and the SFG phenotype is noticeable for all features. Of these, the gaslike cell distribution and the spheroid surface area exhibit especially high distances. Conversely, the high deviation score of the spheroids with collagen concentration 2 mg/ml to the S phenotype stems mostly from the difference in surface deformation, and is less pronounced in the other features. Regarding the lowest observed deviation scores, collagen density 4 spheroids and “deformed” spheroids match comparatively well for each feature, and the spheroid surface area distance between them is the lowest of all. Collagen concentration 2 mg/ml and “disordered” spheroids also exhibit a low deviation score between each other, due to low difference in cell density distribution, Voronoi cell volume distribution and spheroid surface area.

## Simulated versus experimental spheroid comparison



**Fig 8. Deviation score comparison between *in vitro* spheroids grown in media at four different collagen concentrations  $c$  (data provided by Kang et al [13]), and *in silico* spheroids exhibiting four different phenotypes, simulated by us. a)** Shown are the deviation scores between three replicates of each collagen concentration, grown for three days, and five replicates of each simulated phenotype, simulated for 250 000 MC steps. A single MC step corresponds to roughly 1 s of real time in this context. Each individual deviation score is shown on the left, and the average within a pairing of collagen concentration and phenotype is shown on the right. A top-down view of the spheroid point cloud of a representative replicate for each collagen concentration / phenotype is shown next to both heatmaps. Blue cells are classified as non-gaslike, and red cells are classified as gaslike. See S2 Fig for an enlarged view of all experimental spheroid point clouds, and Fig 6 for an enlarged view of all simulated spheroid point clouds. We observe the highest average deviation scores between  $c = 2$  mg/ml and the “spherical” phenotype. The lowest average deviation score is found between  $c = 4$  mg/ml and the “deformed” phenotype **b)** Individual metric distances for each of the features constituting the overall deviation score. Shown are the standardized and weighted metric distances for each of the features constituting the overall deviation score. Shown are the standardized and weighted metric distances for each collagen concentration, grown for three days, and five replicates of each simulated phenotype, simulated for 250 000 MC steps. A single MC step corresponds to roughly 1 s of real time in this context. Due to the overall deviation score being a sum of all weighted feature distances, the color range has been adjusted here. The highest deviation score observed in a) is a combination of high metric distances in all features, especially the spheroid surface deformation. On the other hand, the lowest deviation score observed in a) stems from overall low values, especially in the spheroid surface area.

<https://doi.org/10.1371/journal.pcbi.1010471.g008>

## 2.9 Nastjapy

During our derivation of the overall deviation score and its application to various data, we developed the Python package *Nastjapy*. Through this, we wanted to facilitate the use of our procedure by others. The package can be found at <http://www.gitlab.com/nastja/nastjapy>. *Nastjapy* allows the investigation of spheroids and other single-cell resolved data from different origins. It thereby unifies the analysis pipeline for simulated data and data from multiple experimental sources. See Section 4.4 for more details.

## 3 Discussion and conclusions

Both experimentalists and theorists produce data concerning tumor spheroids. However, both the quantitative comparison between different experiments or simulations, and processes such as fitting simulations to experimental data, are hindered by the lack of an adaptable distance measure that captures the similarity of the spatial features of two spheroids. We aimed to solve this issue via the following steps. First, we proposed a set of five relevant spatial features, which could be extracted from spheroid point clouds. Next, we devised metrics to compare each feature, and combined all metrics into an overall deviation score. We also provided an optimization scheme which could be used to adapt the deviation score to the specific use case. For this, we turned to four *in silico* spheroid phenotypes which emerged from our simulations, and used them to standardize and combine the metrics into the overall deviation score  $D_{ij}$ . We did this by weighing individual metrics differently while maximizing a phenotype separation property (see Eq 7). We characterized the behavior of our features by applying four different transformations to a point cloud obtained from a spherical simulated spheroid. We were able to confirm rotational invariance by analyzing the “rotation” transformation, and monotony for the others. While the features related to the spheroid surface showed some instabilities for higher transformation strengths in the “noise” and “deformation” transformations, this only occurred when the point cloud was so disordered that a solid core could no longer be defined. Overall, the behavior of the features was therefore considered suitable to quantitatively compare the structure of spheroids. Interestingly, the overall deviation score did not only scale monotonously with the strength of the studied transformations, but did so approximately linearly within the domains of interest. This is a useful property for a distance measure, which was not used as a constraint, but instead emerged as a result of our optimization scheme.

During the investigation of our simulated spheroid phenotypes, we found that our deviation score distinguished well between dissimilar spheroid phenotypes, as all phenotypes showed the highest deviation score towards the “disordered” phenotype (see Fig 6). This large distance was a value that was desired, as this phenotype did not have a solid core and was therefore deemed to be least similar to all other phenotypes. The slightly lower deviation score found between the “disordered” and “deformed” phenotypes can be attributed to the fact that our spheroid surface extraction method constructs surfaces for each cluster of cells that is close enough together to be considered solidlike. These surfaces sum up to a large overall surface area, and are highly deformed due to their irregular structure. These features are therefore less meaningful if a singular bulk structure is not present. On the other hand, even glioblastoma spheroids, which originate from one of the most invasive types of cancer, have been shown to retain a bulk structure [32]. We therefore view the “disordered” phenotype as an edge case, but wanted to show it for completeness. Since we observed significant deviation between the other three phenotypes (see Fig 6b) and S1 Fig, we were able to confirm that our strategy is also applicable to more similar spheroids. We therefore consider  $D_{ij}$  to represent a useful metric for the systematic quantification of spheroid similarity. This was further confirmed by our analysis of experimentally measured spheroids generated by Kang et al. [13].

During our analysis of the experimental spheroids, we observed that statistically significant difference between the four collagen concentrations only occurred after three days of growth (see Fig 7c). This suggests that there exists a minimum time which is required until differences in the spheroid structure manifest themselves sufficiently to be detected by *Nastjapy*. On the other hand, this could also be used as a tool to study such timeframes in more detail, e.g. when investigating the response time of drug or radiation treatments.

One of the possible applications of the deviation score is to use it as a reliable objective function for fitting simulated spheroids to experimental data. Therefore, we included a comparison between simulations with unfitted, default parameters, and the aforementioned experimental spheroid data. Here, we were able to highlight, which features contributed most towards each deviation score, and to show that the quantities matched well with a visual comparison. Since the “disordered” phenotype is not present in the experimental data, we expected to see high deviation scores in the top rows of Fig 8a). However, this was not the case, which once again stems from how the surface extraction method functions when a solid spheroid bulk is not present. While we still consider the “disordered” phenotype as an edge-case for spheroids, this represents an opportunity to further develop these features in the future. Distinguishing between singular and multiple bulk structures could become relevant once we apply *Nastjapy* to the analysis of more complex tissues, e.g. organoids.

The features we have defined here admittedly have some limitations. They cannot, for example, measure the dynamics of cell movement over time. Also, spheroids or other tissues we might want to apply this method to, may be composed of multiple different types of cells, and we currently do not distinguish between these. However, since we implemented our strategy in our freely available *Nastjapy* framework (see Section 4.4), it can easily be extended. We aim to further develop this in the future, via incorporating more features. Points of interest would be generating features spanning multiple timesteps, e.g. cell velocity correlation and autocorrelation. Furthermore, we envision features such as the distribution of different cell types, which will enable the application of the analysis scheme to the aforementioned non-spheroid tissue models.

## 4 Materials and methods

### 4.1 Model description

*Cells in Silico* is a framework for simulating the dynamics of cells and tissues at subcellular resolution, which was previously developed by our group [21]. It combines a Cellular Potts Model (CPM) at the microscale with nutrient and signal exchange at the mesoscale and an agent-based layer at the macroscale. This enables detailed capture of individual cell dynamics. Furthermore, as an extension of the *NASTJA* framework [33] its efficiency scales excellently with increasing system size and CPU core number. Hence, *CiS* has already been used for simulating tissues composing millions of cells [21]. Here, we briefly outline the main properties of the microscale, mesoscale and macroscale layers, and a more detailed description can be found in [21].

**Microscale.** The CPM was developed by Graner and Glazier in 1992 [34], as an extension of the Potts model. In it, a system of lattice points on a regular grid is propagated according to its overall energy. Cells are defined as aggregates of points of the same type (see S5 Fig), and the overall energy of the system is built of multiple components  $E_i$ , which dictate the morphology of and interaction between the cells. Weighted by coupling factors  $\lambda_i$ , they are combined

into the following Hamiltonian:

$$\begin{aligned}
 H_{\text{CPM}} &= \sum_i \lambda_i E_i \\
 &= \lambda_v \sum_{c \in C} (v(c) - V(\tau(c)))^2 && \text{Cell volumes} \\
 &+ \lambda_s \sum_{c \in C} (s(c) - S(\tau(c)))^2 && \text{Cell surfaces} \\
 &+ \sum_{i \in \omega} \sum_{j \in N(i)} A_{\tau(c_i), \tau(c_j)} (1 - \delta(c_i, c_j)) && \text{Cell-cell adhesion}
 \end{aligned} \tag{14}$$

where  $c$  is a cell from the set of all cells  $C$ ,  $\tau(c)$  is the type of cell  $c$ ,  $s(c)$  and  $v(c)$  are the current surface and volume of cell  $c$ ,  $S(\tau)$  and  $V(\tau)$  are the target surface and volumes of cells of type  $\tau$ ,  $A$  is the adhesion coefficient matrix for all cell types,  $N(i)$  are all lattice points neighboring point  $i$ , and  $\delta$  is the Kronecker delta. Eq 14 can be extended to include further effects, such as cell motility [35] (see also Section 4.2).

**Mesoscale.** CiS includes the capability of introducing signals or nutrients to the system. These can be exchanged between cells via the cell-cell interface. As this functionality is outside of the scope of this study, we only briefly mention it here and refer the reader to [21].

**Macroscale.** While using the CPM layer allows for excellent reproduction of cell shape and deformation, there are other important cellular functions which are not intrinsically captured. For example, the CPM Hamiltonian does not in itself include the effect of cell division. Furthermore, while self-propelled cell motility can be added to Eq 14 [35], the direction of the motility vector must be periodically updated for each cell, to ensure realistic movement, e.g. via random walk (see also Section 4.2). This requires information on the cell center location, which must be extracted from the CPM. A third aspect, which is very important for simulating realistic tumors, is the capability of *in silico* cell mutation. Here, cell parameters such as division rate, motility magnitude, cell-cell adhesion etc. must be adjusted at the time of division. All the aforementioned aspects are treated in the macroscale layer. It combines information gathered from the lower layers with higher-level parameters, which results in an agent-based system. Here, the conditions for cell division are checked, the division process is carried out, the motility direction is updated, etc.

By combining micro-, meso- and macroscale, we gain a versatile tool, which can then be parameterized.

## 4.2 Model parameters

As mentioned in section 2.1, we simulated a multitude of spheroids using CiS. We based our simulations on experimental spheroid data provided by our collaborators [13]. Hence, each simulated spheroid had an initial diameter of 200  $\mu\text{m}$ , contained roughly 2000 cells, and was placed in the center of a volume spanning 800 x 800 x 800  $\mu\text{m}^3$ . Using CiS, we propagated this system at a range of different simulation parameter combinations, which are highlighted below.

**Extracellular matrix.** The extracellular matrix (ECM) is a scaffold within tissues, which connects cells and serves both as a structural component and cell maintenance network [36]. It is composed of overlapping fibrous polymers, such as collagens, proteoglycans and glycoproteins [37]. Tumor spheroids are often placed into a collagen matrix, which serves as a proxy for an *in vivo* ECM [13, 38]. To capture this in our simulations, we modeled the ECM as overlapping, randomly oriented fibers, which were placed within the system volume surrounding the spheroid (see Fig 1a). These fibers represented rigid obstacles for the cells, to which they

could adhere, but which could not be displaced. Since in reality the ECM is not a solid structure, but can be modified and degraded by cells [39], this alone was an overly simplified description. We therefore added a degradation effect, by which cells removed ECM lattice points with which they were in direct contact on the CPM lattice. This occurred after a set number of MC steps, as described by the ECM degradation period parameter. During each degradation event, a lattice point in contact with a cell was removed with a probability of 50%. For our simulations, we varied both the ECM density and the ECM degradation period. The ECM density was varied between qualitative values of 729 and 1728 fibers. The ECM degradation was either disabled, or its period was varied between 1 000 and 5 000 MC steps. Finally, we wanted to investigate the effect of ECM alignment. Since it is known that tumors remodel their ECM, and ECM alignment is one of the main drivers of invasion [24], we decided to include both an unaligned and a radially aligned ECM in our studies (see Fig 1a). We performed simulations both with aligned and unaligned ECM.

**Cell-cell adhesion.** Changes in the adhesion strength between cells are a well known factor which facilitates invasion [7]. We therefore varied the adhesion strength parameter within our simulations, by changing the adhesion coefficient matrix in the third component of Eq 14. This matrix describes the strength of adhesion between different cell types, as well as the strength of adhesion between cells and the ECM.

**Cell motility magnitude and persistence.** Similar to cell-cell adhesion, the cell motility is strongly connected to the invasion properties of cells [40]. To include it, we modified the CPM Hamiltonian by adding a directional potential to each cell with the following contribution [35]:

$$H_{\text{CPM,mot}} = H_{\text{CPM}} + \lambda_{\text{mot}} \cdot \sum_{c \in C} \vec{m}_c \cdot \vec{R}_c, \quad \text{Cell motility} \quad (15)$$

$$\vec{m}_c = p \cdot (\vec{R}_c(t) - \vec{R}_c(t - \Delta t)) + (1 - p) \cdot \vec{\eta}$$

where  $\vec{m}_c$  is the potential and direction a cell  $c$  experiences,  $R_c$  is the center of mass of cell  $c$ , and  $\vec{\eta}$  is a random vector obtained by a Wiener process [41]. The motility is implemented as a modified persistent random walk of each cell. The energy contribution of each cell is the dot product of  $\vec{R}_c$  and  $\vec{m}_c$ , which in turn is determined by the cell's previous movement and the random vector  $\vec{\eta}$ . The mixture of persistent and random movement can be chosen by the persistence parameter  $p \in [0, 1]$ . Cells with  $p = 0$  perform purely random walks, and cells with  $p = 1$  perform purely persistent walks. The coupling strength of this energy term to the CPM is given by  $\lambda_{\text{mot}}$ .

**Cell division.** Including the effect of cell proliferation is crucial when simulating growing tissues. In CiS, this is implemented as follows: upon division, half of those lattice points on the CPM layer which belong to the dividing cell are assigned a new value, corresponding to the ID of the new cell. The old cell is kept, but its surface, volume, age and generation are adjusted, such that both cells resemble daughter cells of the original one. A division condition needs to be fulfilled in order for a cell to divide. This condition is checked at every MC step. It can be customized, and can include nutrient availability, cell volume, division probability, and cell generation, i.e. the maximum number of divisions per original cell. Since we did not explicitly model nutrient distribution in this study, we focused on the last three aspects. Hence, in our simulations a cell's volume had to be at least 90% of its target volume. Furthermore, in the experimental data by Kang et al [13], the cell number roughly doubles after three days of growth. Therefore, we set the division probability per MC step such that the overall cell number would double after 250 000 MC steps. Finally, we set the cell generation condition such that each cell could only divide once.

### 4.3 Experimental spheroid preparation and analysis

All experimental methods were previously reported by Kang et al. [13] and are briefly summarized here. Multicellular tumor spheroids were formed by seeding highly invasive, triple-negative MDA-MB-231 breast cancer cells [42] in low-attachment 96 well plates (Corning, No. 07201680) in the presence of 2.5% v/v Matrigel (Corning, No. 354234) [43]. Using this approach, ~1000 cancer cells coalesced into a spherical aggregate (i.e., a tumor spheroid) of 300-to-400 $\mu\text{m}$  in diameter over the course of 48 hours. Once formed, individual spheroids were fully embedded into a 3D fibrous gel prepared using rat-tail collagen I (Corning, No. 354249) [44]. As shown in Kang et al. [13], by varying the collagen concentration between 1 and 4 mg/ml, one can tune the fiber density and overall mechanical properties of the collagen network surrounding each tumor spheroids. MDA-MB-231 spheroids were then cultured in such 3D micro-environments for either 1 hour (Day 0), 24 hours (Day 1), 48 hours (Day 2), or 72 hours (Day 3). While at Day 0 all cells remained within the main spheroid (solid-like phase), over the course of 3 days tumor spheroids progressively developed strikingly different patterns of invasion as a function of collagen concentration, including single cell invasion in 1–2 mg/ml collagen (gas-like phase) and collective invasion in 3–4 mg/ml collagen (liquid-like invasion) [13]. For each time point, spheroids were fixed, optically cleared [45], stained with DAPI (Fisher Scientific, No. D1306), and imaged using a Bruker Ultima Investigator multi-photon microscope equipped with a long working distance 16x water-immersion objective (Nikon, 0.8 N.A., 3mm working distance) to enable whole-spheroid imaging [13]. The 3D positions of DAPI-stained cell nuclei were finally identified using a custom Matlab code developed by Kang et al. [13] and used herein as point cloud data to extract features from experimental spheroids. In this work we used point cloud data from spheroids imaged at days 0–1–2–3 in collagen concentrations of 1–2–3–4 mg/ml,  $n = 3$  per group, except for day 0 in 1 mg/ml ( $n = 2$ ), and day 2 in 2 mg/ml ( $n = 9$ ).

### 4.4 Nastjapy

To facilitate the use of the analysis pipeline presented in this study, we developed the Python package *Nastjapy*, which can be found at <http://www.gitlab.com/nastja/nastjapy>. For this study, *Nastjapy* served three functions:

1. providing a unified interface for processing data from multiple different sources
2. performing efficient and parallel feature extraction and analysis
3. adapting and computing the deviation score for specific applications.

We have implemented multiple loading functions into *Nastjapy*, such that it can treat spheroids and other single-cell resolved data from different sources. It currently supports loading point cloud data in the CSV, HDF5, SQLite and matlab file formats. These data are loaded into so-called *DataHandler* objects, which are structured the same way regardless of the origin of the data. Hence, except for the data contained in it, the *DataHandler* object of a simulated spheroid is the same as that of an experimentally measured spheroid. Each *DataHandler* object includes functionalities for visualizing the point cloud data, and extracting the features discussed in section 2.2. Furthermore two or more *DataHandler* objects can be compared with each other. For this, the timesteps to be compared between the spheroids are mapped to each other, and the metric distances are then calculated for each extracted feature. Extracting all features for many different spheroids, possibly at multiple points in time, as well as the comparison between a large number of spheroids, can quickly become computationally expensive. We have therefore added the option to parallelize the feature extraction for individual

*DataHandler* objects on multiple CPU cores via MPI. As we discussed in section 2.4, we have implemented a use case adaptation method, in which the feature distances are standardized, and the weight factors for each feature distance are optimized to maximize the relation stated in Eq 7. All functions necessary to perform this for a set of *DataHandler* objects are implemented in *Nastjapy*.

## Supporting information

**S1 Fig. Deviation score box plots for simulated phenotypes.** a) Each phenotype compared to the “deformed” phenotype. b) Each phenotype compared to the “spherical with far gaslikes” phenotype. c) Each phenotype compared to the “disordered” phenotype. Significance was determined using Welch’s t-test.

(TIF)

**S2 Fig. Top-down view of spheroid point clouds obtained from *in vitro* MDA-MB-231 spheroids grown in collagen media at four different concentrations c.** Spheroids were imaged after one, two and three days of growth, and three replicates were imaged per concentration and growth duration. Blue cells are classified as non-gaslike, and red cells are classified as gaslike.

(TIF)

**S3 Fig. Deviation score box plots of *in vitro* spheroids grown in media at four different collagen concentrations (data provided by Kang et al [13]).** The horizontal axis denotes the growth duration of the spheroids within the respective boxplot, and the vertical axis denotes the reference collagen concentration. Significance was determined using Welch’s t-test.

(TIF)

**S4 Fig. Visualization of the spheroid surface deformation feature on a triangulated spheroid surface.** Shown is the surface, composed of many connected triangles, as well as an example of the normal and origin vectors of two triangle vertices. During the feature extraction, the scalar product between these two vectors is calculated for each vertex in the triangle mesh, and grouped in a histogram. Non-deformed surfaces will contain more vertices in which the two vectors are approximately parallel (left example), while strongly deformed surfaces will contain many vertices in which there is a strong deviation between the two (right example).

(TIF)

**S5 Fig. Sketch of the three layers of Cells in Silico (CiS).** A 3D Cellular Potts Model (CPM, shown in 2D for illustrative purpose) at the microscale is combined with nutrient and signal exchange at the mesoscale and an agent-based layer at the macroscale. This enables detailed capture of individual cell dynamics. Adapted from [46].

(TIF)

## Acknowledgments

The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. ([www.gauss-centre.eu](http://www.gauss-centre.eu)) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre (JSC).

## Author Contributions

**Conceptualization:** Julian Herold, Eric Behle, Jakob Rosenbauer, Jacopo Ferruzzi, Alexander Schug.

**Data curation:** Julian Herold, Eric Behle, Jakob Rosenbauer.

**Formal analysis:** Julian Herold, Eric Behle.

**Funding acquisition:** Alexander Schug.

**Investigation:** Julian Herold, Eric Behle.

**Methodology:** Julian Herold, Eric Behle, Jakob Rosenbauer.

**Project administration:** Alexander Schug.

**Resources:** Jacopo Ferruzzi, Alexander Schug.

**Software:** Julian Herold, Eric Behle, Jakob Rosenbauer.

**Supervision:** Alexander Schug.

**Validation:** Julian Herold, Eric Behle.

**Visualization:** Eric Behle.

**Writing – original draft:** Julian Herold, Eric Behle, Jakob Rosenbauer, Jacopo Ferruzzi, Alexander Schug.

**Writing – review & editing:** Julian Herold, Eric Behle, Jakob Rosenbauer, Jacopo Ferruzzi.

## References

1. The global challenge of cancer. *Nature Cancer*. 2020; 1(1):1–2. <https://doi.org/10.1038/s43018-019-0023-9> PMID: [35121840](https://pubmed.ncbi.nlm.nih.gov/35121840/)
2. Hanahan D. Hallmarks of Cancer: New Dimensions. *Cancer Discovery*. 2022; 12(1):31–46. <https://doi.org/10.1158/2159-8290.CD-21-1059> PMID: [35022204](https://pubmed.ncbi.nlm.nih.gov/35022204/)
3. Pei-Hsun W, Gilkes Daniele M, Phillip Jude M, Akshay N, Wen-Tao CT, Jorge M, et al. Single-cell morphology encodes metastatic potential. *Science Advances*. 2020; 6(4):eaaw6938. <https://doi.org/10.1126/sciadv.aaw6938>
4. Tashireva LA, Zavyalova MV, Savelieva OE, Gerashchenko TS, Kaigorodova EV, Denisov EV, et al. Single Tumor Cells With Epithelial-Like Morphology Are Associated With Breast Cancer Metastasis. *Frontiers in Oncology*. 2020; 10. <https://doi.org/10.3389/fonc.2020.00050> PMID: [32154161](https://pubmed.ncbi.nlm.nih.gov/32154161/)
5. Wang Z, An K, Li R, Liu Q. Tumor Macroscopic Morphology Is an Important Prognostic Factor in Predicting Chemotherapeutic Efficacy and Clinical Outcomes of Patients With Colorectal Neuroendocrine Neoplasms, One Multicenter Retrospective Cohort Study. *Frontiers in Endocrinology*. 2021; 12. <https://doi.org/10.3389/fendo.2021.801741> PMID: [34987482](https://pubmed.ncbi.nlm.nih.gov/34987482/)
6. Hanahan D, Weinberg RA. The Hallmarks of Cancer. *Cell*. 2000; 100(1):57–70. [https://doi.org/10.1016/S0092-8674\(00\)81683-9](https://doi.org/10.1016/S0092-8674(00)81683-9) PMID: [10647931](https://pubmed.ncbi.nlm.nih.gov/10647931/)
7. Klymkowsky MW, Savagner P. Epithelial-mesenchymal transition: a cancer researcher's conceptual friend and foe. *The American journal of pathology*. 2009; 174:1588–93. <https://doi.org/10.2353/ajpath.2009.080545> PMID: [19342369](https://pubmed.ncbi.nlm.nih.gov/19342369/)
8. Montagud A, de Leon MP, Valencia A. Systems biology at the giga-scale: Large multiscale models of complex, heterogeneous multicellular systems. *Current Opinion in Systems Biology*. 2021; 28:100385. <https://doi.org/10.1016/j.coisb.2021.100385>
9. Nia HT, Munn LL, Jain RK. Physical traits of cancer. *Science (New York, NY)*. 2020; 370. <https://doi.org/10.1126/science.aaz0868> PMID: [33122355](https://pubmed.ncbi.nlm.nih.gov/33122355/)
10. Gunti S, Hoke ATK, Vu KP, London NRJ. Organoid and Spheroid Tumor Models: Techniques and Applications. *Cancers*. 2021; 13. <https://doi.org/10.3390/cancers13040874> PMID: [33669619](https://pubmed.ncbi.nlm.nih.gov/33669619/)
11. Perche F, Torchilin VP. Cancer cell spheroids as a model to evaluate chemotherapy protocols. *Cancer biology & therapy*. 2012; 13:1205–13. <https://doi.org/10.4161/cbt.21353>

12. Grosser S, Lippoldt J, Oswald L, Merkel M, Sussman DM, Renner F, et al. Cell and Nucleus Shape as an Indicator of Tissue Fluidity in Carcinoma. *Phys Rev X*. 2021; 11:011033.
13. Kang W, Ferruzzi J, Spatarelu CP, Han YL, Sharma Y, Koehler SA, et al. A novel jamming phase diagram links tumor invasion to non-equilibrium phase separation. *iScience*. 2021; 24(11):103252. <https://doi.org/10.1016/j.isci.2021.103252> PMID: 34755092
14. Swat MH, Thomas GL, Shirinifard A, Clendenon SG, Glazier JA. Emergent Stratification in Solid Tumors Selects for Reduced Cohesion of Tumor Cells: A Multi-Cell, Virtual-Tissue Model of Tumor Evolution Using CompuCell3D. *PLOS ONE*. 2015; 10(6):1–40. <https://doi.org/10.1371/journal.pone.0127972>
15. Goodin DA, Frieboes HB. Simulation of 3D centimeter-scale continuum tumor growth at sub-millimeter resolution via distributed computing. *Computers in biology and medicine*. 2021; 134:104507. <https://doi.org/10.1016/j.combiomed.2021.104507> PMID: 34157612
16. Fu X, Zhao Y, Lopez JL, Rowan A, Au L, Fendler A, et al. Spatial patterns of tumour growth impact clonal diversification in a computational model and the TRACERx Renal study. *Nature Ecology & Evolution*. 2022; 6(1):88–102. <https://doi.org/10.1038/s41559-021-01586-x>
17. Bull JA, Mech F, Quaiser T, Waters SL, Byrne HM. Mathematical modelling reveals cellular dynamics within tumour spheroids. *PLOS Computational Biology*. 2020; 16(8):1–25. <https://doi.org/10.1371/journal.pcbi.1007961> PMID: 32810174
18. Amereh M, Edwards R, Akbari M, Nadler B. In-Silico Modeling of Tumor Spheroid Formation and Growth. *Micromachines*. 2021; 12. <https://doi.org/10.3390/mi12070749> PMID: 34202262
19. Browning AP, Sharp JA, Murphy RJ, Gunasingh G, Lawson B, Burrage K, et al. Quantitative analysis of tumour spheroid structure. *Elife*. 2021; 10:e73020. <https://doi.org/10.7554/eLife.73020> PMID: 34842141
20. Szymańska Z, Skrzeczkowski J, Miasojedow B, Gwiazda P. Bayesian inference of a non-local proliferation model. *Royal Society open science*. 2021; 8(11):211279. <https://doi.org/10.1098/rsos.211279> PMID: 34849247
21. Berghoff M, Rosenbauer J, Hoffmann F, Schug A. Cells in Silico—introducing a high-performance framework for large-scale tissue modeling. *BMC bioinformatics*. 2020; 21(1):1–21. <https://doi.org/10.1186/s12859-020-03728-7> PMID: 33023471
22. Friedl P, Locker J, Sahai E, Segall JE. Classifying collective cancer cell invasion. *Nature Cell Biology*. 2012; 14(8):777–783. <https://doi.org/10.1038/ncb2548> PMID: 22854810
23. Lautscham LA, Kämmerer C, Lange JR, Kolb T, Mark C, Schilling A, et al. Migration in Confined 3D Environments Is Determined by a Combination of Adhesiveness, Nuclear Volume, Contractility, and Cell Stiffness. *Biophysical journal*. 2015; 109:900–13. <https://doi.org/10.1016/j.bpj.2015.07.025> PMID: 26331248
24. Wang WY, Pearson AT, Kutys ML, Choi CK, Wozniak MA, Baker BM, et al. Extracellular matrix alignment dictates the organization of focal adhesions and directs uniaxial cell migration. *APL bioengineering*. 2018; 2:046107. <https://doi.org/10.1063/1.5052239> PMID: 31069329
25. Aurenhammer F. Voronoi Diagrams—a Survey of a Fundamental Geometric Data Structure. *ACM Comput Surv*. 1991; 23(3):345–405. <https://doi.org/10.1145/116873.116880>
26. Lorensen WE, Cline HE. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *SIGGRAPH Comput Graph*. 1987; 21(4):163–169. <https://doi.org/10.1145/37402.37422>
27. Villani C. *Optimal transport: old and new*. vol. 338. Springer; 2009.
28. Levina E, Bickel P. The earth mover’s distance is the mallows distance: Some insights from statistics. In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. vol. 2. IEEE; 2001. p. 251–256.
29. Kursawe J, Baker RE, Fletcher AG. Approximate Bayesian computation reveals the importance of repeated measurements for parameterising cell-based models of growing tissues. *Journal of theoretical biology*. 2018; 443:66–81. <https://doi.org/10.1016/j.jtbi.2018.01.020> PMID: 29391171
30. Khosla P, Teterwak P, Wang C, Sarna A, Tian Y, Isola P, et al. Supervised contrastive learning. *Advances in Neural Information Processing Systems*. 2020; 33:18661–18673.
31. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*. 2020; 17:261–272. <https://doi.org/10.1038/s41592-019-0686-2> PMID: 32015543
32. Stein AM, Demuth T, Mobley D, Berens M, Sander LM. A mathematical model of glioblastoma tumor spheroid invasion in a three-dimensional in vitro experiment. *Biophysical journal*. 2007; 92:356–65. <https://doi.org/10.1529/biophysj.106.093468> PMID: 17040992

33. Berghoff M, Kondov I. Non-collective Scalable Global Network Based on Local Communications. In: 2018 IEEE/ACM 9th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems (scalA); 2018. p. 25–32.
34. Graner Fmc, Glazier JA. Simulation of biological cell sorting using a two-dimensional extended Potts model. *Phys Rev Lett*. 1992; 69:2013–2016. <https://doi.org/10.1103/PhysRevLett.69.2013> PMID: [10046374](https://pubmed.ncbi.nlm.nih.gov/10046374/)
35. Guisoni N, Mazzitello KI, Diambra L. Modeling Active Cell Movement With the Potts Model. *Frontiers in Physics*. 2018; 6. <https://doi.org/10.3389/fphy.2018.00061>
36. Kular JK, Basu S, Sharma RI. The extracellular matrix: Structure, composition, age-related differences, tools for analysis and applications for tissue engineering. *Journal of tissue engineering*. 2014; 5:2041731414557112. <https://doi.org/10.1177/2041731414557112> PMID: [25610589](https://pubmed.ncbi.nlm.nih.gov/25610589/)
37. Bonnans C, Chou J, Werb Z. Remodelling the extracellular matrix in development and disease. *Nature reviews Molecular cell biology*. 2014; 15:786–801. <https://doi.org/10.1038/nrm3904> PMID: [25415508](https://pubmed.ncbi.nlm.nih.gov/25415508/)
38. Mark C, Grundy TJ, Strissel PL, Böhringer D, Grummel N, Gerum R, et al. Collective forces of tumor spheroids in three-dimensional biopolymer networks. *eLife*. 2020; 9:e51912. <https://doi.org/10.7554/eLife.51912> PMID: [32352379](https://pubmed.ncbi.nlm.nih.gov/32352379/)
39. Lu P, Takai K, Weaver VM, Werb Z. Extracellular matrix degradation and remodeling in development and disease. *Cold Spring Harbor perspectives in biology*. 2011; 3. <https://doi.org/10.1101/cshperspect.a005058> PMID: [21917992](https://pubmed.ncbi.nlm.nih.gov/21917992/)
40. Stuelten CH, Parent CA, Montell DJ. Cell motility in cancer invasion and metastasis: insights from simple model organisms. *Nature Reviews Cancer*. 2018; 18(5):296–312. <https://doi.org/10.1038/nrc.2018.15> PMID: [29546880](https://pubmed.ncbi.nlm.nih.gov/29546880/)
41. Durrett R. *Probability: theory and examples*. vol. 49. Cambridge university press; 2019.
42. Cailleau R, Mackay B, Young RK, Reeves WJ. Tissue Culture Studies on Pleural Effusions from Breast Carcinoma Patients. 1974; 34:10.
43. Ivascu A, Kubbies M. Rapid Generation of Single-Tumor Spheroids for High-Throughput Cell Function and Toxicity Analysis. *Journal of Biomolecular Screening*. 2006; 11(8):922–932. <https://doi.org/10.1177/1087057106292763> PMID: [16973921](https://pubmed.ncbi.nlm.nih.gov/16973921/)
44. Harjanto D, Maffei JS, Zaman MH. Quantitative Analysis of the Effect of Cancer Invasiveness and Collagen Concentration on 3D Matrix Remodeling. *PLoS ONE*. 2011; 6(9):e24891. <https://doi.org/10.1371/journal.pone.0024891> PMID: [21980363](https://pubmed.ncbi.nlm.nih.gov/21980363/)
45. Susaki EA, Tainaka K, Perrin D, Yukinaga H, Kuno A, Ueda HR. Advanced CUBIC protocols for whole-brain and whole-body clearing and imaging. *Nature Protocols*. 2015; 10(11):1709–1727. <https://doi.org/10.1038/nprot.2015.085> PMID: [26448360](https://pubmed.ncbi.nlm.nih.gov/26448360/)
46. Rosenbauer J, Berghoff M, Schug A. Emerging Tumor Development by Simulating Single-cell Events. *bioRxiv*. 2020.

# Appendix B

## Embedded Publication 2

Eric Behle, Julian Herold, Alexander Schug, *Leveraging Experimental Vasculature Data for High Resolution Brain Tumor Simulations*, PREPRINT, bioRxiv, 2024

### Copy permissions

This preprint was published in an open-access preprint server with the following copyright remark: *The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.*

### Contributions

For this article, I developed the *mousebrainsimulator* package for processing of mouse brain microscopy data. I wrote the majority of the manuscript, roughly 90%. I extended the code of *Cells in Silico*, and performed all simulations and data analyses. I generated all figures.

### Full article

# Leveraging Experimental Vasculature Data for High Resolution Brain Tumor Simulations

Eric Behle<sup>1</sup>, Julian Herold<sup>2</sup>, Alexander Schug<sup>1,3\*</sup>

<sup>1\*</sup>Jülich Supercomputing Centre, Jülich Research Centre, Wilhelm-Johnen Str. 1, Jülich, 52425, North-Rhine Westphalia, Germany.

<sup>2</sup>Steinbuch Centre for Computing, Karlsruhe Institute of Technology, Kaiserstr. 12, Karlsruhe, 76131, Baden-Württemberg, Germany.

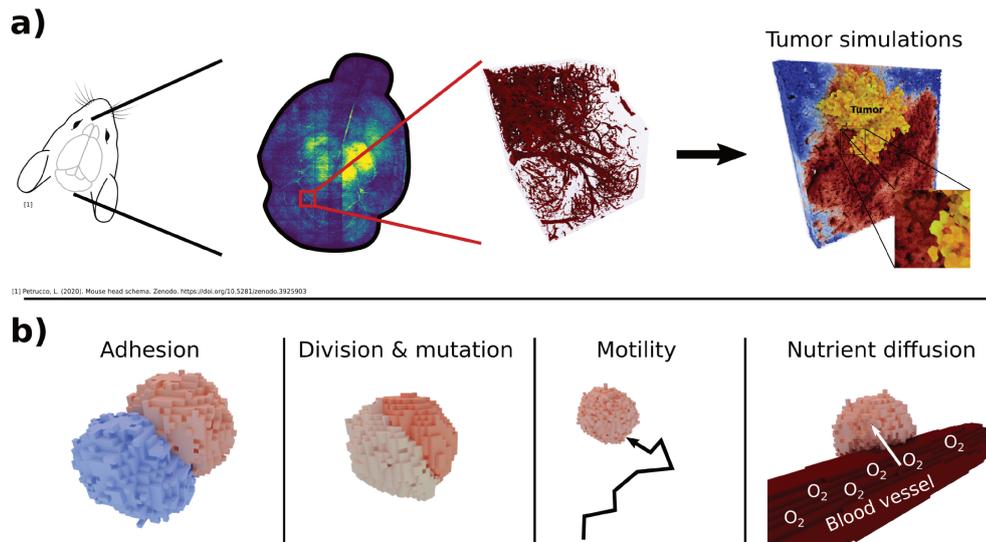
<sup>3\*</sup>Biology, University of Duisburg-Essen, , Essen, 45141, North-Rhine Westphalia, Germany.

\*Corresponding author(s). E-mail(s): [al.schug@fz-juelich.de](mailto:al.schug@fz-juelich.de);  
Contributing authors: [e.behle@fz-juelich.de](mailto:e.behle@fz-juelich.de); [julian.herold@kit.edu](mailto:julian.herold@kit.edu);

## Abstract

Cancer remains a leading cause of mortality. Multidisciplinary studies probe its complex pathology to increase treatment options. Computational modeling of tumor growth on high-performance computing resources offers microscopic insight into its progress and a valuable avenue for advancing our understanding. However, the effective initialization and parameterization of the underlying models require high-resolution data from real tissue structures. Here, we leveraged high performance computing resources and a massive dataset of a mouse brain's entire vascular network. We processed these image stacks into detailed three-dimensional representations, identified brain regions of interest, and conducted a series of large-scale simulations to investigate how tumor growth is influenced by local vascular network characteristics. By simulating tumor growth with sub-cellular resolution, we can probe to which extent vessel density and vessel network length influence tumor growth. We determined that vessel density is the primary determinant of growth rate. Finally, our results allowed us to extrapolate tumor cell growth predictions for the entire mouse brain, highlighting the critical role of vascular topology in tumor progression. Such increasingly realistic simulations of cancer cells and their microenvironment enable researchers to increasingly bridge the gap between basic biology and clinical practice, ultimately supporting the development of more effective personalized cancer therapies.

**Keywords:** Tumor simulation, mouse brain, vasculature



**Fig. 1** Visual abstract. Mouse brain vasculature microscopy data are a useful resource for the initialization of tumor growth simulations. We utilized data by Di Giovanna et al. [1], who imaged the vasculature network of an entire mouse brain and provided light-sheet microscopy z-stacks. In this study, we implemented a pipeline to clean, align and combine these stacks, obtaining three-dimensional representations. We then performed an analysis of the network topology at different positions and used this analysis to identify areas of interest within the brain. We then performed a large scale analysis of tumor growth simulations depending on the nutrient environment based on these regions of interest.

## 1 Introduction

Cancer remains one of the leading causes of death worldwide. Thus, it poses an ever-present global health challenge that urgently requires improved therapeutic strategies and deeper insights into its pathology. Despite remarkable advances in cancer treatment, the complexity of tumor growth, its interaction with surrounding tissues, and its eventual resistance to therapies often lead to challenges in effective clinical management. Central to addressing these challenges are advanced simulations of cancer tissue. The massive growth of compute resources allows studies which offer the possibility of highly detailed, sub-cellular insights into the dynamics driving tumor progression. They also enable researchers to visualize and analyze critical interactions within the tumor microenvironment, that are otherwise challenging to study *in vivo*. Such interactions include nutrient gradients, vascular influences, and cellular displacement. By mimicking these intricate biological processes in a controlled computational setting, simulations have emerged as a vital pillar in cancer research. They bridge the gap between empirical experimental data and theoretical models, complementing wet-lab findings and aiding in hypothesis testing. [2–5]. Computational simulations have been instrumental not only in cancer studies but also in fields such as developmental biology,

where they provide insights into tissue mechanics and cellular processes in embryogenesis and wound healing. This multidisciplinary approach facilitates the development of comprehensive models that can inform various stages of cancer development and therapeutic response. These can vary in resolution and scope, from modeling cellular and molecular dynamics at a fine scale to examining tissue-wide phenomena [6, 7]. The ultimate goal of many research initiatives is to develop “digital twins” of tumors: highly realistic, computational representations of tumors within healthy tissue that mimic their real-life counterparts [8–10]. These digital twins promise to revolutionize personalized medicine, enabling simulations of disease progression and treatment response specific to an individual’s unique biological profile. Building such a model requires an accurate and dynamic morphology of both the healthy and tumorous tissue. It needs to capture the growth behavior of tumors as they interact with the cellular environment, migrate, and displace other cells. A key factor in these simulations is nutrient availability, as nutrient gradients largely dictate tumor growth dynamics and progression patterns [11, 12]. The transition from avascular (limited to nutrient diffusion) to vascularized growth phases is a significant turning point in tumor development, where angiogenesis, the formation of new blood vessels, provides the tumor with an essential nutrient supply. Without this transition, tumors cannot grow beyond a critical volume of around 1-3 mm<sup>3</sup> before nutrient limitations halt further expansion. Therefore, the study of nutrient supply mechanisms and vascularization processes is a focal point in cancer research, with numerous studies investigating how nutrient gradients impact tumor viability and aggressiveness [13–15]. Previous simulations relied on idealized representations of nutrient distributions, such as simple radial gradients that assume uniformity in nutrient availability [16]. However, these models lack the complexity of real tissue environments, where nutrient distribution is influenced by an intricate vascular network. A more realistic depiction of the tissue microenvironment requires detailed information about vascular topology, particularly at the capillary level, which is essential for understanding how tumors respond to spatially heterogeneous nutrient supplies. Here, we aim to go beyond idealized models by integrating data that closely approximates actual tissue environments. Leveraging recent advances in fluorescent microscopy and image processing, we utilize high-resolution data from Di Giovanna et al. [1], which provides a capillary-scale map of the entire vascular network within a mouse brain. Such data allow us to incorporate highly realistic tissue morphology into our simulations, capturing the nuanced interactions between tumor cells and their vascular surroundings. Our approach takes advantage of computational resources and simulation frameworks. High-performance computing clusters, now more accessible than ever, support these complex models by handling vast datasets and performing simulations at sub-cellular resolution. A key factor in executing these simulations is the parallelization and scalability of simulation software; the CiS (Cells in Silico) framework [17], used in our study, demonstrates effective performance scaling across thousands of computing nodes, making it a suitable tool for handling the demands of high-resolution, three-dimensional simulations.

In the following, we will first outline the dataset, describing the imaging techniques and data processing methods used to achieve a detailed three-dimensional representation of vascular topology. We will then discuss the integration of these vascular maps

into our simulation pipeline and present our analysis of how variations in vascular network characteristics influence tumor growth patterns. A critical question in tumor growth is the dependence of growth on the local microenvironment and accessibility of nutrients. Here, large scale simulations provide sub-cellular insight and identify key factors catalyzing growth. Finally, we will summarize the insights gained from these simulations and propose directions for future research, including potential applications in the development of patient-specific cancer therapies.

## 2 Results

### 2.1 Vascular Data Processing for Large-Scale Simulations

To inform our simulations based on the vascular network topology, we first processed the raw data into analyzable structures. Here, we detail this processing pipeline.

#### 2.1.1 Raw Microscopy data

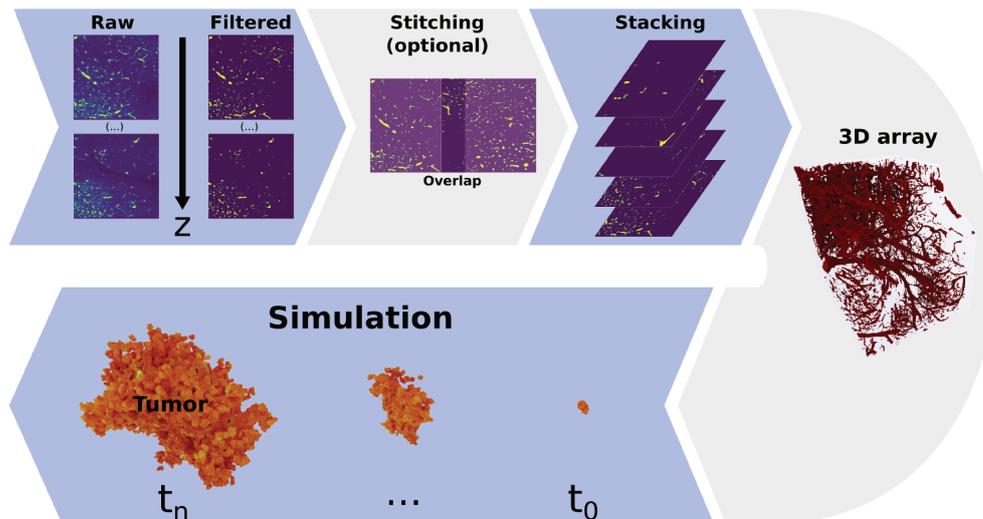
Di Giovanna et al. measured the entire vascular network of a mouse brain at capillary resolution [1] (see Figure 1 and 2). The raw data generated by them are available in the form of z-stacks of light sheet microscopy images [1]. Each stack contains 2160 microscopy images with a resolution of  $2048 \times 2048$  px. One px represents an area of  $0.65 \times 0.65 \mu\text{m}^2$ , and the spacing between each image is  $2 \mu\text{m}$  in z-direction. Hence, a single stack depicts a volume of  $1331 \times 1331 \times 4320 \mu\text{m}^3$ . Furthermore, adjacent stacks have an overlap of roughly  $300 \mu\text{m}$  in x- and y-direction.

#### 2.1.2 Processing into 3D volumes

Processing the data into a usable form presents several challenges. First, since they are raw microscopy images, they contain noise. Secondly, there may be artifacts from the experimental procedure, especially at the edges of each stack. Therefore, the first step in obtaining a three-dimensional representation of all stacks was to denoise them. Towards this goal, we built binary masks from the microscopy images of each of them. This was done by applying a threshold filter on each image. All images were then combined in a three-dimensional array. The resulting array contains the value 1 in locations where a blood vessel was present, and the value 0 otherwise. Next, we rescaled this three-dimensional binary mask. By doing so we matched the native resolution of our model, such that 1 voxel represents a volume of  $1 \mu\text{m}^3$ . This was done for each individual microscopy stack. Finally, in order to obtain a roughly cubic geometry for our simulations, we split each stack into four substacks along the z-axis. The final volume of the binary mask of each substack was therefore  $V = 1331 \times 1331 \times 1080 \mu\text{m}^3$ .

#### 2.1.3 Stitching multiple stacks

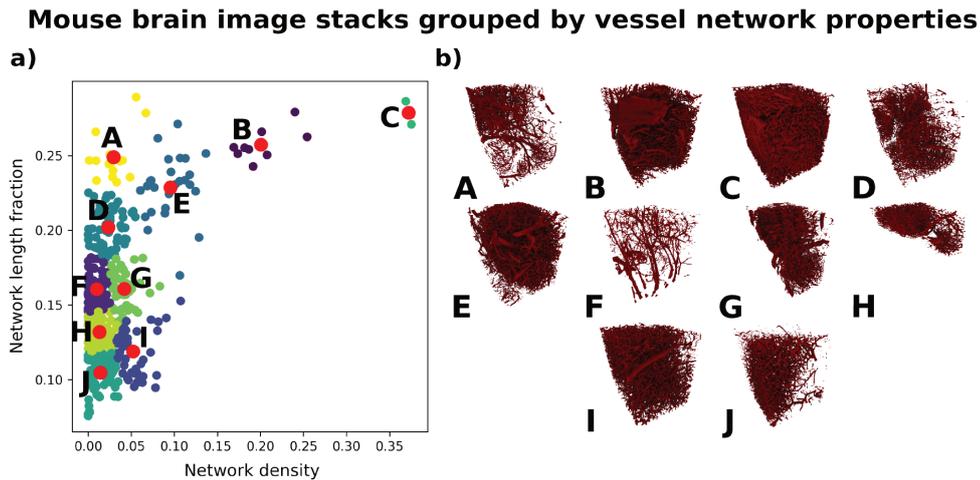
We implemented a method that enables us to stitch neighboring regions into a single stack. This addresses the second challenge of the raw data by minimizing artifacts at the stack edges. See SI appendix A for more details.



**Fig. 2** Overview of the mouse brain vasculature data processing pipeline. The pipeline begins with raw microscopy images, which are converted into binary masks using a threshold filter to isolate vascular structures. Due to the size limitation of individual z-stacks ( $1331 \times 1331 \times 4320 \mu\text{m}^3$ ), a stitching procedure can be applied to extend the simulation volume. By optimizing the alignment of adjacent z-stacks, we minimize artifacts in the merged image. These binary masks are then assembled into a 3D array, which is subsequently imported into the *Cells in Silico* (CiS) platform, providing spatial and structural information for nutrient source modeling in our simulations.

#### 2.1.4 Microscopy stack grouping

Even though supercomputers have grown immensely in computational power in recent years [18], simulating the entire 3D representation of the mouse brain at  $\mu\text{m}$ -resolution would still be a monumental challenge. On the other hand, it could be argued that due to the respective similarity between different regions (see also SI Figure 5), significant insights can already be gained by focusing on a subset of archetypes. To determine these archetypes, we decided to characterize the substacks and find points of interest. We used two parameters for this characterization: the vessel network density  $\rho$ , and a parameter which we call the network length fraction  $f_1$ .  $\rho$  was taken as the fraction of vessel voxels within the substack.  $f_1$  was calculated through a skeletonizing procedure [19]. Here, each binary mask was eroded, reducing the diameter of each vessel to 1 voxel.  $f_1$  is obtained by computing the fraction of remaining to originally present vessel voxels. A value of  $f_1$  close to 0 indicates a low number of thick blood vessels and hence a low overall network length. A value close to 1 indicates many thin vessels which increases the overall network length.  $f_1$  can also be viewed as a measure of the relative surface area of the vessel network. As seen in Figure 3 a), k-means clustering by  $\rho$  and  $f_1$  does not yield fully distinct clusters, except for B and C. Nonetheless, it enables us to choose groups of interest much better than randomly picking from all substacks. For the following simulation study, we chose the substacks which were



**Fig. 3** Detection of points of interest within mouse brain vasculature data. **a)** Processed mouse brain microscopy stacks arranged regarding their local blood vessel network properties. Shown are, for each stack, the network density  $\rho$ , and the network length fraction  $f_l$ . The color indicates the cluster membership obtained from kmeans-clustering. The red points represent the centroid of each cluster. **b)** Snapshot of the three-dimensional structure of the stacks chosen for further study. Each chosen stack is the one closest to the centroid of each cluster.

closest to the cluster centroids (red points in Figure 3 a)). A snapshot of each chosen stack's three-dimensional structure is also shown in Figure 3 b).

## 2.2 Influence of vessel network topology on tumor growth

After choosing areas of interest within the mouse brain, we investigated the influence of  $\rho$  and  $f_l$  on tumor growth. Here, we wanted to test two non-exclusive hypotheses regarding nutrient driven tumor growth:

- Tumor growth is mostly driven by the density of the vascular network ( $\rho$ ). In this case the local blood vessels drive tumor growth, since higher density leads to more available nutrients.
- Tumor growth is mostly driven by the local topology of the vascular network ( $f_l$ ). In this case different local topologies result in differences of the vessel network surface area, influencing the available amount of nutrients.

The simplest assumption was that there is a linear relationship between  $\rho$  and the final tumor cell count. To evaluate this, we performed simulations of tumors seeded in different parts of the brain using the *Cells in Silico* (CiS) tissue simulation framework [17]. See section 4.1 for a detailed summary of CiS.

### 2.2.1 Simulations

In each simulation, blood vessels were represented by inserting specialized voxels into the Cellular Potts Model (CPM) layer of the CiS framework (see 4.3). These vessel voxels, derived from the binary mask data from section 2.1.2, served as nutrient

sources. The remaining volume was populated with healthy tissue cells, maintaining a density consistent with mouse brain cell populations ( $\sim 130,000$  cells) [20]. A single initial tumor, consisting of approximately 10 cells, was then seeded.

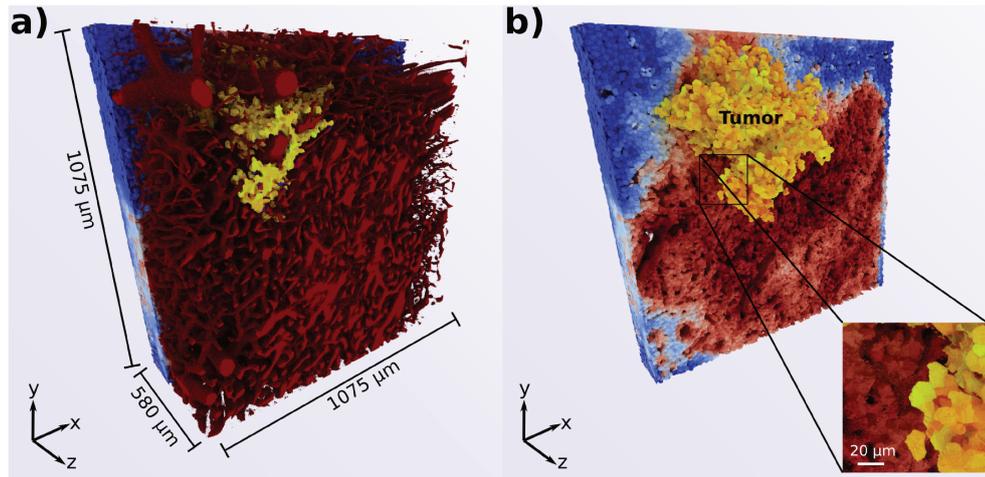
Nutrient diffusion was modeled using the CiS compound diffusion layer (see section 4.1), with a single nutrient acting as a proxy for oxygen, glucose, and other essential compounds delivered by blood vessels. Tumor cells could proliferate if they met the necessary nutrient threshold. With each division, a tumor cell could undergo a mutation affecting either its cell-cell adhesion strength or its motility. Additionally, each cell consumed nutrients, and tumor cells would die if their nutrient levels dropped below a critical threshold.

It is important to note that the traditional CPM, as used here, does not fully capture the intricate morphologies of neurons and glial cells. Nonetheless, CPMs are widely applied in similar studies for tissue modeling [21, 22]. We argue that this simplification is justified, as the primary cell-cell interactions driving these simulations — adhesion forces and nutrient diffusion — are closely related to the shared surface area between cells. Introducing more complex cell shapes would mainly result in two effects: (a) increasing shared surface area for cells already in contact, effectively modifying adhesion and diffusion parameters for all cells uniformly, and (b) creating small shared surfaces between cells previously not in contact. Since the first effect would scale uniformly and the second effect introduces minimal interaction, the impact of these adjustments would be limited. Moreover, adding complex shapes would significantly increase both model and computational complexity, with only marginal gains in accuracy.

More details on the model parameters are provided in section 4.2 and in SI table 1. We performed eight simulations for each centroid substack obtained in section 2.1.4. In each simulation, the tumor was seeded at one of eight points. These points were at one third and two thirds of each simulation box diagonal. Each simulation ran for a total amount of 250 000 MC steps. A snapshot of a single simulation after 100 000 Monte Carlo (MC) steps is provided in Figure 4.

### 2.2.2 Tumor growth behavior

To compare all simulations, we first calculated the local blood vessel properties in the vicinity of each tumor. For this we chose a sphere of radius  $250 \mu\text{m}$  around the initial tumor center. We then calculated the local density  $\rho_{\text{local}}$  and local network length fraction  $f_{1,\text{local}}$  within this sphere. For the calculation of  $f_{1,\text{local}}$  we had to further refine our procedure, since we noted artifacts which distorted the skeletonization. After applying a post-processing pipeline of morphological operations, this was improved. Finally, we determined the number of tumor cells at the end of each simulation and compared the quantities in Figure 5. While the relationship between the network length fraction and the final tumor cell count is less clear, the dependence of the final tumor cell count on the local blood vessel density shown in Figure 5 b) roughly follows a logistic curve:



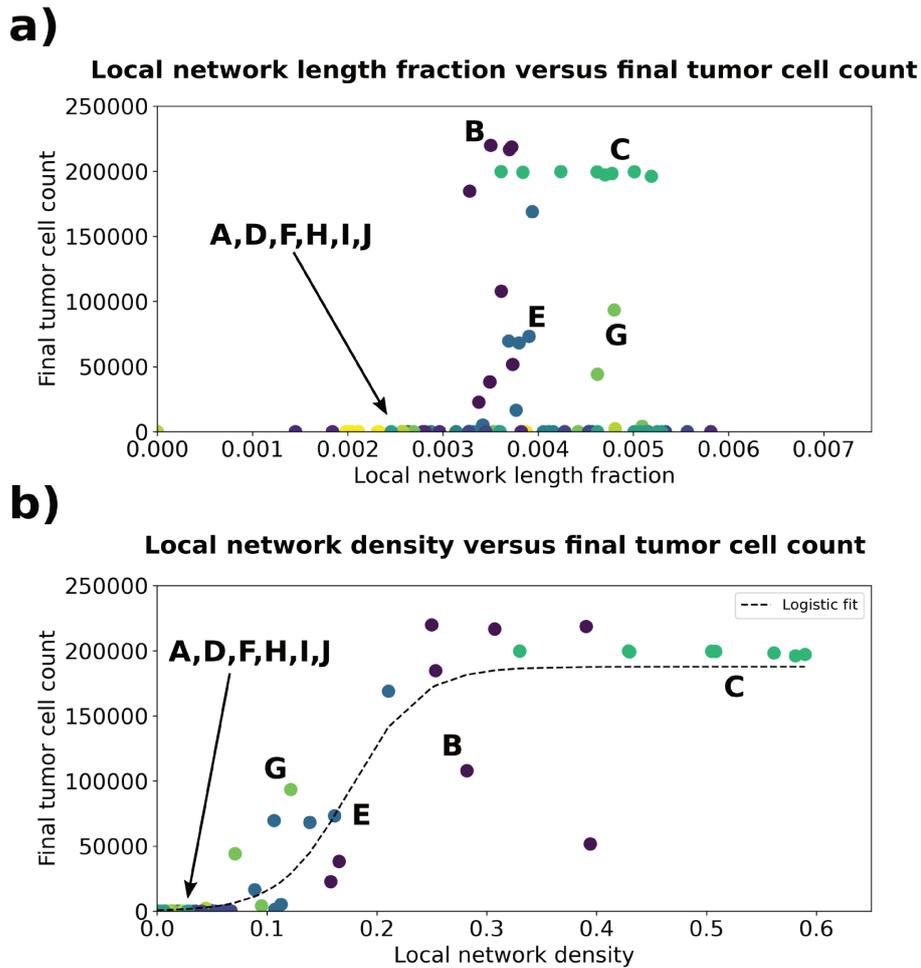
**Fig. 4** Visualization of a simulated tumor growing within a mouse brain vasculature environment (cluster B in Fig. 3b) after 100,000 Monte Carlo (MC) steps. **(a)** The dark red structure represents the blood vessel network, providing nutrient sources. **(b)** The tumor cells (yellow) are shown expanding in three dimensions, with the surrounding healthy cells color-coded according to their nutrient content (blue to red gradient), indicating the spatial distribution of nutrient availability. The displayed volume is a subset of the entire simulation space ( $1344 \times 1344 \times 1080 \mu\text{m}^3$ ) to focus on the tumor's immediate environment.

$$N(\rho) = A \cdot \frac{e^{B \cdot \rho}}{e^{B \cdot \rho} + C}. \quad (1)$$

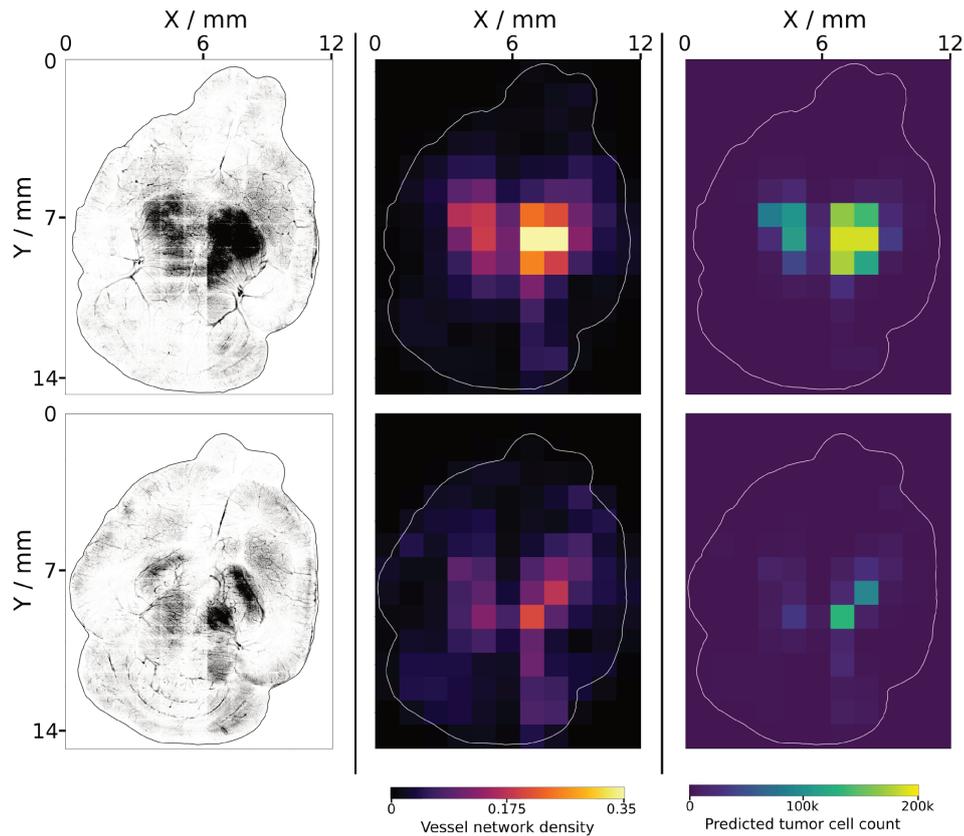
These data show that the simple assumption of a linear dependence between vessel density and tumor growth rate does not hold. Instead there exists a threshold above which the tumor grows rapidly. Furthermore, the higher blood vessel surface area resulting from the high network length fractions does not appear to be a main driver for tumor growth.

### 2.2.3 Predicted tumor growth in the entire mouse brain

We can use the relation between local blood vessel density and final tumor cell count to estimate the tumor size for the entire mouse brain by fitting equation 1 to the data shown in Figure 5 b) (see also SI table 2). We used the resulting values to calculate the expected cell count for the given vessel density  $\rho$  of each substack. Within our simulation environment tumors are expected to grow largest in the densest regions in the center of the brain, and to not grow well in the periphery (see Figure 6 and SI Figure 3).



**Fig. 5** Mousebrain vasculature property comparison **a)** Network length fraction versus final tumor cell count. Shown are the color-coded final tumor cell counts for eight simulations of each respective microscopy stack chosen in section 2.1.4. In each simulation, the tumor was placed at a different starting position, and hence the local network length fraction differs. The network length fraction serves as a proxy for the vessel network length. We see no clear dependence of the final tumor cell count on the network length. **b)** Local blood vessel density versus final tumor cell count. Shown are the color-coded final tumor cell count for eight simulations of each respective microscopy stack chosen in section 2.1.4. In each simulation, the tumor was placed at a different starting position, and hence the local blood vessel density differs. The growth behavior follows a logistic curve.



**Fig. 6** Mouse brain blood vessel density and predicted tumor growth. Left: Top-down views of the processed and stitched microscopy stacks for the lowest two layers of the brain. Middle: heatmaps showing the blood vessel volume fraction for each microscopy stack. Right: heatmaps showing the predicted tumor cell count for each microscopy stack. Predicted values were obtained using equation 1 with the values from fitting to the simulation data (see section 2.2.3 and SI table 2).

### 3 Discussion and conclusion

Computational studies of tumor growth can profit immensely from existing experimental data. The challenge is to incorporate these data in a meaningful manner. In this study, we showed how we utilized highly detailed microscopy data on mouse brain vasculature in order to inform our simulation framework CiS. After developing a data processing pipeline for the microscopy stacks found in the raw data, we characterized each processed stack by its vessel network topology. We utilized two parameters for this characterization: the blood vessel density  $\rho$ , and the network length fraction  $f_1$ . After choosing ten substacks of interest based on the two parameters, we performed simulations of tumor growth in this wide range of nutrient environments. Placing the initial tumor in different locations yielded different local starting topologies, and we observed different growth behavior based on this. We saw that the main driver of

tumor growth is the local vessel density, and that the final cell count roughly follows a logistic curve. While there are some outliers in Figure 5 b), these show that the final tumor cell count is not determined solely by the starting position of the tumor, but also its surrounding further from it (see also SI Figure 4).

The network length fraction, which approximates the distribution of vessel thicknesses, does not appear to have significant influence on the final tumor size. This was surprising to us, since we expected that a higher network length fraction would correspond to a larger surface area of the vessels, leading to increased nutrient diffusion.

Using a logistic curve fitted to the local density data, we predicted the tumor growth in the entire brain. Strong tumor growth was predicted mainly in the dense regions in the center of the lower two planes of the brain. One major caveat to mention here is that we used the global vessel density per stack for the prediction, and did not look at more localized vessel properties. This effectively represents an averaging of larger volumes, thus losing information about inhomogeneities. Hence, we might miss some regions which locally surpass the established threshold. Furthermore, as seen in the left pane of Figure 6, there are some residual brightness artifacts from the microscopy, which may influence the results.

This study shows that there still is a large amount of information to be gained from the existing experimental data on tumors. We aim to utilize these data for further studies of more realistic tumors. In particular, the stitching procedure that we mentioned in section 2.1 could be used to simulate tumors of multiple  $\text{mm}^3$  in volume. Such system sizes would enable us to perform *in silico* radio- and chemotherapy treatment studies in the future. Furthermore, it is a step on the way to simulating fully vascularized macroscopic tumor growth with our model. Our future studies will focus on implementing tumor-induced angiogenesis, the remaining step on this path.

## 4 Materials and Methods

### 4.1 Model description

*Cells in Silico* was developed by our group as a framework for simulating the dynamics of cells and tissues at subcellular resolution [16, 17, 23]. It is a hybrid model composed of a Cellular Potts Model (CPM), nutrient and signal exchange functionalities, and an agent-based layer. With it, we can capture individual cell dynamics in detail. Furthermore, it is an extension of the *NAStJA* framework [24]. Being specifically designed for deployment in High Performance Computing environments, its efficient scaling behaviour allows the simulations of even macroscopic tissues [17, 25]. Hence, CiS has already been used for simulating tissues composing millions of cells [17]. Here, we briefly outline the main properties of the individual layers. A detailed description can be found in [17].

### CPM layer

The CPM is an extension of the Potts model, developed by Graner and Glazier [26]. In it, we describe cells as connected voxels of same value on a three-dimensional regular grid. The value of individual voxels may change over time depending on the energy of the system. This energy is described by the following Hamiltonian:

$$\begin{aligned}
 H_{\text{CPM}} &= \sum_i \lambda_i E_i \\
 &= \lambda_V \sum_{c \in C} \left( v(c) - V(\tau(c)) \right)^2 && \text{Cell volumes} \\
 &+ \lambda_S \sum_{c \in C} \left( s(c) - S(\tau(c)) \right)^2 && \text{Cell surfaces} \\
 &+ \sum_{i \in \omega} \sum_{j \in N(i)} A_{\tau(c_i), \tau(c_j)} \left( 1 - \delta(c_i, c_j) \right) && \text{Cell-cell adhesion}
 \end{aligned} \tag{2}$$

where  $\lambda_i$  is the weight factor for the energy contribution  $E_i$ ,  $c$  is a cell from the set of all cells  $C$ ,  $\tau(c)$  is the type of cell  $c$ ,  $s(c)$  and  $v(c)$  are the current surface and volume of cell  $c$ ,  $S(\tau)$  and  $V(\tau)$  are the target surface and volumes of cells of type  $\tau$ ,  $A$  is the adhesion coefficient matrix for all cell types,  $N(i)$  are all lattice points neighboring point  $i$ , and  $\delta$  is the Kronecker delta. Equation 2 can be extended to include further effects, such as cell motility [27]. In order to propagate the system, we utilize the Metropolis algorithm [28]. First, we change the values of randomly picked voxels into those of a respective neighboring voxel and calculate the energy difference  $\Delta E$  between the old and new system state:

$$\Delta E = H_{\text{CPM, new}} - H_{\text{CPM, old}} \tag{3}$$

We then decide whether to accept or reject the change. The acceptance probability  $p_{\text{accept}}$  is calculated as follows:

$$p_{\text{accept}} = \begin{cases} 1, & \text{if } \Delta E \leq 0. \\ e^{-\frac{\Delta E}{T}}, & \text{otherwise.} \end{cases} \tag{4}$$

Here,  $T$  is the temperature of the system. Upon successive propagation steps, cells will expand, compress, deform and move, thereby mimicking their behavior in real tissue. CiS also includes the possibility to add "solid" voxels to the system. Such voxels cannot change their value, and no other voxels can assume a value associated with a solid. This enables us to add structures such as fixed blood vessels.

### Compound-exchange layer.

In addition to detailed cellular structure, CiS contains functionalities for adding compounds such as signals or nutrients to the system. These are exchanged between cells via the cell-cell interface. Solids on the CPM layer can act as sources, thereby mimicking nutrient supplying blood vessels. The flux  $J_{i,j}^k$  of compound  $k$  between two cells  $i$  and  $j$  is calculated as:

$$J_{i,j}^k = D_{\tau(i),\tau(j)} \left( \frac{A_{i,j}}{A_i} + \frac{A_{i,j}}{A_j} \right) ([S]_j^k - [S]_i^k) \quad (5)$$

where  $D_{\tau(i),\tau(j)}$  is the diffusion constant between cells of type  $\tau(i)$  and  $\tau(j)$ ,  $A_{i,j}$  is the interface area between cell  $i$  and cell  $j$ ,  $A_i$  and  $A_j$  are the overall surface areas of cell  $i$  and cell  $j$ , and  $[S]_i^k$  and  $[S]_j^k$  are the compound concentrations within each cell. The aforementioned solids in the CPM layer can also function as compound sources or sinks. The impact of the compound on the system is defined on the agent based layer.

### ***Agent based layer (ABL).***

On the ABL, each cell within the tissue is tracked as an individual agent. The properties of each agent are adjusted based on information from the other two layers, and their effects are calculated. Furthermore, cellular functions that are not intrinsically captured by the CPM or the diffusion model, such as cell division and cell death, are implemented here. The use of an ABL also allows us to include cell mutation into the framework: upon each cell division, the properties of the two daughter cells can be changed, such that new cell types emerge.

By combining all three layers, we gain a versatile tool, which can then be parameterized to describe a wide range of system dynamics.

## **4.2 Model parameters**

CiS contains many parameters which influence the behavior of the simulated tissue. Here we highlight those used in this study.

### ***Cell-cell adhesion***

Adhesion interactions between cells are known to vary within tissues, and especially within tumors. They mediate tissue fluidity and effects such as the epithelial-to-mesenchymal transition (EMT) demonstrate their importance [29]. In our simulations, healthy cells adhere to each other and to tumor cells at a strength of 50 AU, which we define as the baseline adhesion. Tumor cells can have an adhesion strength between 0 and 140 AU (see *Cell mutation* paragraph below).

### ***Cell motility magnitude and persistence***

The CPM Hamiltonian of CiS includes the possibility of adding a cell motility term [27]:

$$H_{\text{CPM, mot}} = H_{\text{CPM}} + \lambda_{\text{mot}} \cdot \sum_{c \in \mathcal{C}} \vec{m}_c \cdot \vec{R}_c, \quad \text{Cell motility} \quad (6)$$

$$\vec{m}_c = p \cdot \left( \vec{R}_c(t) - \vec{R}_c(t - \Delta t) \right) + (1 - p) \cdot \vec{\eta}$$

Here  $\vec{m}_c$  represents the potential and direction of a cell  $c$ ,  $R_c(t)$  and  $R_c(\Delta t)$  represent the center of mass of cell  $c$  at MC step  $t$  and  $\Delta t$ , respectively, and  $\vec{\eta}$  is a random

vector obtained by a Wiener process [30]. Each cell follows a modified persistent random walk. The energy contribution of each cell is the dot product of  $\vec{R}_c$  and  $\vec{m}_c$ .  $\vec{m}_c$  is obtained using the cell's previous movement and the random vector  $\vec{\eta}$ . By varying the persistence parameter  $p \in [0, 1]$ , we can mediate the contribution of persistent and random movement. At  $p = 0$  a cell performs a purely random walk, while at  $p = 1$  it performs a purely persistent walk [31]. Finally, we mediate the coupling strength of the motility term to the CPM via  $\lambda_{\text{mot}}$ . In our simulations, healthy cells have a motility strength of  $\lambda_{\text{mot,healthy}} = 0$  AU. The motility of tumor cells varies between 0 and 105 AU (see *Cell mutation* paragraph below).

### *Nutrient content*

Cells require nutrients to proliferate, and a growing tumor is strongly limited by the nutrients supplied by the surrounding vasculature. We therefore defined a "nutrient" compound on the diffusion layer. For simplicity, this single nutrient combines oxygen and all other nutrients supplied by blood vessels. It is released from solids on the CPM layer to adjacent cells. The solids act as a reservoir with a constant concentration of 4 AU. Therefore, the nutrient amount of all cells lies between 0 and 4 AU. The diffusion constant between vessels and cells, and between any two cells is  $D = 0.1$  AU, and diffusion takes place once every 10 MC steps. Finally, each cell consumes nutrients: healthy cells use 0.05 AU nutrients every 10 MC steps. We assume that tumor cells are less efficient and therefore use 0.1 AU nutrients every 10 MC steps [32].

### *Cell division rate*

Cell agents within CiS are capable of cell division. When this occurs, the cell is split across a randomly oriented division plane on the CPM layer. Half of the voxels belonging to the old cell are assigned a new value which corresponds to the ID of the new cell. The volume, surface, age and generation properties of the old cell agent are then adjusted, until finally both cells resemble daughter cells of the original one. In order for division to occur, a division condition must be fulfilled. This condition is customizable and can include a number of different terms, e.g. oxygen content, cell volume, cell generation and a random component. Since the healthy tissue in our simulations are meant to model a coarse-grained tumor environment, only the tumor cells are capable of division. Our assumption here is that the surrounding tissue is in a steady-state of division and death. In order for a tumor cell to divide, its volume must be at least 60% of its target volume, and its nutrient content must be greater than or equal to 1 AU. Furthermore, we set the division probability such that a cell divides roughly every 10000 MC steps.

### *Cell mutation*

Upon division, a cell can mutate, thereby changing some of its agent properties. We decided to focus on two commonly used properties: the cell motility and adhesion strength [33]. A cell has a 10% chance to increase or decrease either its motility or adhesion strength upon division. Since we did not see significant influence of the mutation properties on the simulations, we only briefly mention this here.

### *Cell death*

While cell agents are capable of division, they can also die. In real tissues cells can die due to multiple reasons, but for simplicity here we decided to only focus on the nutrient availability. Hence, a cell will die if its nutrients drop below 0.1 AU.

## 4.3 Loading of blood vessel data into CiS

CiS is parallelized using the Message Passing Interface (MPI), and therefore workers have separate memory address spaces [17]. Each worker is used to simulate a sub part of the overall CPM volume, which we call a block. We have implemented the loading of the blood vessel data such that each worker loads its individual file at the start of the simulation. These files are generated prior to running the simulation. We first specify the number and size of blocks in each dimension, and split the blood vessel binary mask into accordingly sized sub volumes. These sub volumes are then written out as files named *n.raw*, with *n* being the MPI rank to which the sub volume belongs. Each file contains a list of voxel values, which is the flattened version of the respective sub volume. The flattening is done in C-order (row major).

## 4.4 Data visualization

The visualization of the three-dimensional binary masks and the simulations presented a challenge, since non-specialized data visualization frameworks have difficulty to do so at this scale (which are up to giga or even tera-voxel). It was achieved using volcanite (to be published) based on previous work [34].

## Acknowledgements

The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. ([www.gauss-centre.eu](http://www.gauss-centre.eu)) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre (JSC). We thank Max Piochowiak for insightful discussions. We are grateful for funding by the Helmholtz Association in the programs Helmholtz AI (AS, JH) and HIDSS4Health (AS, JK). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

## Code and data availability

The code of CiS can be found at <https://gitlab.com/nastja/nastja>. The data and analysis scripts required for reproducing our analysis can be found at <https://gitlab.jsc.fz-juelich.de/behle2/mousebrainvasculaturepaper>. Due to the size of the entire simulation data (roughly 1.1 TB), we only include the last output file of each simulation within this repository. The full data are available upon reasonable request.

## Declarations

The authors declare that they have no competing interests.

## References

- [1] Giovanna, D., Paolo, A., Tibo, A., Silvestri, L., Müllenbroich, M.C., Costantini, I., Allegra Mascaro, A.L., Sacconi, L., Frascioni, P., Pavone, F.S.: Whole-brain vasculature reconstruction at the single capillary level. *Scientific Reports* **8**(1), 12573 (2018)
- [2] Tang, L., Van De Ven, A.L., Guo, D., Andasari, V., Cristini, V., Li, K.C., Zhou, X.: Computational modeling of 3d tumor growth and angiogenesis for chemotherapy evaluation. *PloS one* **9**(1), 83962 (2014)
- [3] Oden, J.T., Lima, E.A., Almeida, R.C., Feng, Y., Rylander, M.N., Fuentes, D., Faghihi, D., Rahman, M.M., DeWitt, M., Gadde, M., *et al.*: Toward predictive multiscale modeling of vascular tumor growth: computational and experimental oncology for tumor prediction. *Archives of Computational Methods in Engineering* **23**, 735–779 (2016)
- [4] Szabó, A., Merks, R.M.: Cellular potts modeling of tumor growth, tumor invasion, and tumor evolution. *Frontiers in oncology* **3**, 87 (2013)
- [5] Runser, S., Vetter, R., Iber, D.: Simucell3d: three-dimensional simulation of tissue mechanics with cell polarization. *Nature Computational Science*, 1–11 (2024)
- [6] Brodland, G.W.: How computational models can help unlock biological systems. *Seminars in Cell & Developmental Biology* **47-48**, 62–73 (2015) <https://doi.org/10.1016/j.semcdb.2015.07.001> . Coding and non-coding RNAs & Mammalian development
- [7] Szymańska, Z., Cytowski, M., Mitchell, E., Macnamara, C.K., Chaplain, M.A.: Computational modelling of cancer development and growth: modelling at multiple scales and multiscale modelling. *Bulletin of Mathematical Biology* **80**(5), 1366–1403 (2018)
- [8] Laubenbacher, R., Adler, F., An, G., Castiglione, F., Eubank, S., Fonseca, L.L., Glazier, J., Helikar, T., Jett-Tilton, M., Kirschner, D., Macklin, P., Mehrad, B., Moore, B., Pasour, V., Shmulevich, I., Smith, A., Voigt, I., Yankeelov, T.E., Ziemssen, T.: Toward mechanistic medical digital twins: some use cases in immunology. *Frontiers in Digital Health* **6** (2024) <https://doi.org/10.3389/fgth.2024.1349595>
- [9] Anvari, S., Nambiar, S., Pang, J., Maftoon, N.: Computational models and simulations of cancer metastasis. *Archives of Computational Methods in Engineering*, 1–23 (2021)

- [10] Zangooui, M.H., Margolis, R., Hoyt, K.: Multiscale computational modeling of cancer growth using features derived from microct images. *Scientific reports* **11**(1), 18524 (2021)
- [11] Chaplain, M.A.J.: Avascular growth, angiogenesis and vascular growth in solid tumours: The mathematical modelling of the stages of tumour development. *Mathematical and Computer Modelling* **23**(6), 47–87 (1996) [https://doi.org/10.1016/0895-7177\(96\)00019-2](https://doi.org/10.1016/0895-7177(96)00019-2)
- [12] Gupta, M.K., Qin, R.-Y.: Mechanism and its regulation of tumor-induced angiogenesis. *World journal of gastroenterology* **9**, 1144–55 (2003)
- [13] Bresch, D., Colin, T., Grenier, E., Ribba, B., Saut, O.: Computational modeling of solid tumor growth: the avascular stage. *SIAM Journal on Scientific Computing* **32**(4), 2321–2344 (2010)
- [14] Sciumè, G., Gray, W., Ferrari, M., Decuzzi, P., Schrefler, B.: On computational modeling in tumor growth. *Archives of Computational Methods in Engineering* **20**, 327–352 (2013)
- [15] Romano, E., Rosa, I., Fioretto, B.S., Manetti, M.: Recent insights into cellular and molecular mechanisms of defective angiogenesis in systemic sclerosis. *Biomedicines* **12**(6), 1331 (2024)
- [16] Rosenbauer, J., Berghoff, M., Glazier, J.A., Schug, A.: Multiscale modeling of spheroid tumors: Effect of nutrient availability on tumor evolution. *J. Phys. Chem. B* **127**(16), 3607–3615 (2023) <https://doi.org/10.1021/acs.jpccb.2c08114>
- [17] Berghoff, M., Rosenbauer, J., Hoffmann, F., Schug, A.: Cells in silico—introducing a high-performance framework for large-scale tissue modeling. *BMC bioinformatics* **21**(1), 1–21 (2020)
- [18] Kesselheim, S., Herten, A., Krajsek, K., Ebert, J., Jitsev, J., Cherti, M., Langguth, M., Gong, B., Stadtler, S., Mozaffari, A., *et al.*: Jewels booster—a supercomputer for large-scale ai research. In: *High Performance Computing: ISC High Performance Digital 2021 International Workshops, Frankfurt Am Main, Germany, June 24–July 2, 2021, Revised Selected Papers 36*, pp. 453–468 (2021). Springer
- [19] Lee, T.C., Kashyap, R.L., Chu, C.N.: Building skeleton models via 3-d medial surface axis thinning algorithms. *CVGIP: Graphical Models and Image Processing* **56**(6), 462–478 (1994) <https://doi.org/10.1006/cgip.1994.1042>
- [20] Keller, D., Erö, C., Markram, H.: Cell densities in the mouse brain: A systematic review. *Frontiers in neuroanatomy* **12**, 83 (2018)
- [21] Giniünaitė, R., Baker, R.E., Kulesa, P.M., Maini, P.K.: Modelling collective cell

- migration: neural crest as a model paradigm. *Journal of Mathematical Biology* **80**(1), 481–504 (2020)
- [22] Buttenschön, A., Edelstein-Keshet, L.: Bridging from single to collective cell migration: A review of models and links to experiments. *PLoS computational biology* **16**(12), 1008411 (2020)
- [23] Herold, J., Behle, E., Rosenbauer, J., Ferruzzi, J., Schug, A.: Development of a scoring function for comparing simulated and experimental tumor spheroids. *PLOS Computational Biology* **19**(3), 1–26 (2023) <https://doi.org/10.1371/journal.pcbi.1010471>
- [24] Berghoff, M., Kondov, I.: Non-collective scalable global network based on local communications. In: 2018 IEEE/ACM 9th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems (scalA), pp. 25–32 (2018). <https://doi.org/10.1109/ScalA.2018.00007>
- [25] Herten, A., Achilles, S., Alvarez, D., Badwaik, J., Behle, E., Bode, M., Breuer, T., Caviedes-Voullième, D., Cherti, M., Dabah, A., Sayed, S.E., Frings, W., Gonzalez-Nicolas, A., Gregory, E.B., Mood, K.H., Hater, T., Jitsev, J., John, C.M., Meinke, J.H., Meyer, C.I., Mezentsev, P., Mirus, J.-O., Nassyr, S., Penke, C., Römmer, M., Sinha, U., St. Vieth, B., Stein, O., Suarez, E., Willsch, D., Zhukov, I.: Application-Driven Exascale: The JUPITER Benchmark Suite (2024). <https://arxiv.org/abs/2408.17211>
- [26] Graner, F.m.c., Glazier, J.A.: Simulation of biological cell sorting using a two-dimensional extended potts model. *Phys. Rev. Lett.* **69**, 2013–2016 (1992) <https://doi.org/10.1103/PhysRevLett.69.2013>
- [27] Guisoni, N., Mazzitello, K.I., Diambra, L.: Modeling active cell movement with the potts model. *Frontiers in Physics* **6** (2018) <https://doi.org/10.3389/fphy.2018.00061>
- [28] Borkar, V.S.: Equation of state calculations by fast computing machines. *Resonance* **27**, 1263–1269 (1953)
- [29] Debnath, P., Huirem, R.S., Dutta, P., Palchaudhuri, S.: Epithelial-mesenchymal transition and its transcription factors. *Bioscience reports* **42** (2022)
- [30] Durrett, R.: *Probability: Theory and Examples*, 5th edn. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge (2019)
- [31] Selmecki, D., Mosler, S., Hagedorn, P.H., Larsen, N.B., Flyvbjerg, H.: Cell motility as persistent random motion: Theories from experiments. *Biophysical Journal* **89**(2), 912–931 (2005) <https://doi.org/10.1529/biophysj.105.061150>

- [32] Warburg, O.: On the origin of cancer cells. *Science* **123**(3191), 309–314 (1956) <https://doi.org/10.1126/science.123.3191.309>  
<https://www.science.org/doi/pdf/10.1126/science.123.3191.309>
- [33] Kalluri, R., Weinberg, R.A.: The basics of epithelial-mesenchymal transition. *The Journal of clinical investigation* **119**, 1420–8 (2009)
- [34] Piochowiak, M., Dachsbacher, C.: Fast compressed segmentation volumes for scientific visualization. *IEEE Transactions on Visualization and Computer Graphics* **30**(1), 12–22 (2024) <https://doi.org/10.1109/TVCG.2023.3326573>

# Appendix C

## Embedded Publication 3

Andreas Herten, Sebastian Achilles, Damian Alvarez, Jayesh Badwaik, **Eric Behle**, Mathis Bode, Thomas Breuer, Daniel Caviedes-Voullième, Mehdi Cherti, Adel Dabah, Salem El Sayed, Wolfgang Frings, Ana Gonzalez-Nicolas, Eric B. Gregory, Kaveh Haghighi Mood, Thorsten Hater, Jenia Jitsev, Chelsea Maria John, Jan H. Meinke, Catrin I. Meyer, Pavel Mezentsev, Jan-Oliver Mirus, Stepan Nassyr, Carolin Penke, Manoel Römmer, Ujjwal Sinha, Benedikt von St. Vieth, Olaf Stein, Estela Suarez, Dennis Willsch, Ilya Zhukov, *Application-Driven Exascale: The JUPITER Benchmark Suite*, SC24, 2024

### Copy permissions

I have not included the final version of the paper [107] due to copyright considerations. Instead I have included the preprint, which was published in an open-access preprint server with the following copyright remark: *CC BY 4.0: Creative Commons Attribution. This license allows reusers to distribute, remix, adapt, and build upon the material in any medium or format, so long as attribution is given to the creator. The license allows for commercial use.*

### Contributions

For this publication, I developed the *NASTJA* part of the benchmark, and performed all simulations and data analyses related to it. I contributed section *IV.A.f* of the manuscript and took part in its revision.

### Full article

# Application-Driven Exascale: The JUPITER Benchmark Suite

Andreas Herten , Sebastian Achilles , Damian Alvarez , Jayesh Badwaik , Eric Behle , Mathis Bode , Thomas Breuer , Daniel Caviedes-Voullième , Mehdi Cherti , Adel Dabah , Salem El Sayed , Wolfgang Frings , Ana Gonzalez-Nicolas , Eric B. Gregory , Kaveh Haghighi Mood , Thorsten Hater , Jenia Jitsev , Chelsea Maria John , Jan H. Meinke , Catrin I. Meyer , Pavel Mezentsev , Jan-Oliver Mirus , Stepan Nassyr , Carolin Penke , Manoel Römmer , Ujjwal Sinha , Benedikt von St. Vieth , Olaf Stein , Estela Suarez , Dennis Willsch , Ilya Zhukov 

Jülich Supercomputing Centre  
Forschungszentrum Jülich  
Jülich, Germany

**Abstract**—Benchmarks are essential in the design of modern HPC installations, as they define key aspects of system components. Beyond synthetic workloads, it is crucial to include real applications that represent user requirements into benchmark suites, to guarantee high usability and widespread adoption of a new system. Given the significant investments in leadership-class supercomputers of the exascale era, this is even more important and necessitates alignment with a vision of Open Science and reproducibility. In this work, we present the JUPITER Benchmark Suite, which incorporates 16 applications from various domains. It was designed for and used in the procurement of JUPITER, the first European exascale supercomputer. We identify requirements and challenges and outline the project and software infrastructure setup. We provide descriptions and scalability studies of selected applications and a set of key takeaways. The JUPITER Benchmark Suite is released as open source software with this work at [github.com/FZJ-JSC/jubench](https://github.com/FZJ-JSC/jubench).

**Index Terms**—Benchmark, Procurement, Exascale, System Design, System Architecture, GPU, Accelerator

## I. INTRODUCTION

The field of High Performance Computing (HPC) is governed by the interplay of capability and demand driving each other forward. During the design and purchase phase of supercomputer procurements for public research, the capability of a machine is usually assessed not only by theoretical, system-inherent numbers, but also by effective numbers relating to actual workloads. These workloads are traditionally benchmark programs that test specific aspects of the system design — like the floating-point throughput, memory bandwidth, or internode latency. While these *synthetic benchmarks* are well-suited for the assessment of distinct features, for a more integrated and realistic perspective, they should be complemented by *application benchmarks*. Application benchmarks use state-of-the-art scientific applications to assess the performance of integrated designs. Complex application profiles utilize various types of hardware resources dynamically during the benchmark’s runtime, showcasing real-world strengths and limitations of the system.

This paper introduces the *JUPITER Benchmark Suite*, a comprehensive collection of 23 benchmark programs meticulously documented and designed to support the procurement of JUPITER, Europe’s first exascale supercomputer. On top of 7 synthetic benchmarks, 16 application benchmarks were developed in close collaboration with domain scientists to ensure relevance and rigor. Additionally, this paper offers valuable insights into the state-of-the-practice of exascale procurement, shedding light on the challenges and methodologies involved.

Preparations for the procurement of JUPITER were launched in early 2022 and finally came to fruition with the awarding of the contract in October 2023. JUPITER is funded by the EuroHPC Joint Undertaking (50%), Germany’s Federal Ministry for Education and Research (25%), and the Ministry of Culture and Science of the State of North Rhine-Westphalia of Germany (25%), and is hosted at Jülich Supercomputing Centre (JSC) of Forschungszentrum Jülich. As part of the procurement, the benchmark suite was developed to motivate the system design and evaluate the proposals committed for the Request for Proposals. The suite focuses on application benchmarks to ensure high practical usability of the system. This work presents the *JUPITER Benchmark Suite* in detail, highlighting design choices and project setup, describing the benchmark workloads, and releasing them as open source software. The suite includes 23 benchmarks across different domains, each with unique characteristics such as compute-intensive, memory-intensive, and I/O-intensive workloads. The applications are grouped into three categories: Base, representing a mixed base workload for the system, High-Scaling, highlighting scalability to the full exascale system, and synthetic, determining various key hardware design features. The benchmark suite represents a first step towards *Continuous Benchmarking* to detect system anomalies during the production phase of JUPITER.

The main contributions of this paper are:

- An in-depth description of the use of benchmarks in HPC procurement, including relevant background information.

- The description of a novel methodology to assess exascale system designs in the form of *High-Scaling benchmarks*.
- The development of suitable benchmark workloads based on a representative set of scientific problems, applications, and synthetic codes.
- Scalability results on the preparation system for all application benchmarks of the suite.
- Insights and best practices learned from application scaling and the procurement process in general.
- Release of the full JUPITER Benchmark Suite as open source software.

After providing background information (section II) and details on the used infrastructure (section III), the suite’s individual benchmarks are presented in section IV. Key takeaways are provided in section V, followed by a conclusion and outlook in section VI.

## II. BACKGROUND

### A. Requirements

The main requirements of the benchmark suite stem from the need to represent existing and upcoming user communities in the system design process. This ensures a fitting design and fosters adoption of the system by users. The suite must cover the wide user portfolio of the HPC center, containing typical applications from various domains utilizing the current HPC infrastructure, and also represent expected future workloads. Moreover, it is essential to ensure diversity in terms of methods, programming languages, and execution models, since such diversity is an inherent characteristic of the application portfolio for upcoming large-scale systems.

The context of a procurement poses high requirements for *replicability*, *reproducibility*, and *reusability* [1]. Replicability, i.e., the seamless execution on the same hardware by the developer, is an elementary requirement to guarantee robustness. Beyond that, reproducibility describes the seamless execution on different hardware by someone else, making it a key requirement in the context of the procurement since both the site and the system provider must be able to run the suite to obtain the same results. Ensuring reusability, i.e., designing the framework for easy adaptation for a variety of tasks in the future, is essential to justify the substantial investment involved in the creation of the suite.

Vendors participating in the procurement process must also invest considerable time in system-specific adjustments while meeting the procurement’s requirements and complying to its rules, usually within short time scales. Therefore, it is in everyone’s best interest to have clear, well-defined guidelines and, ideally, to leverage an existing benchmark suite to build on established expertise.

### B. JUPITER Procurement Scheme

The procurement for the JUPITER system uses a Total-Cost-of-Ownership-based (TCO) value-for-money approach, in which the number of executed reference workloads over the lifespan of the system determines the value. Given the size of state-of-the-art supercomputers as well as their corresponding

power consumption, costs for electricity and cooling are a substantial part of the overall project budget. Using a mixture of application benchmarks as well as synthetic benchmarks, operational costs are computed in a well-defined procedure. While synthetic benchmarks allow for assessing key performance characteristics, they do not allow a realistic assessment of resource consumption during the lifetime of the system for TCO. Therefore, a greater emphasis is placed on application performance rather than on synthetic tests.

Given the targeted system performance of 1 EFLOP/s with 64 bit precision, an additional novel benchmark type focusing on the scale of the system is introduced — *High-Scaling benchmarks*. A subset of applications able to fully utilize JUPITER is identified and use cases were defined to make them part of this novel category. In the course of the paper, we will refer to either *High-Scaling benchmarks* or *Base benchmarks*, to differentiate between both.

The JUPITER system is envisioned to consist of two connected sub-systems, *modules* in the Modular Supercomputing Architecture (MSA) concept [2]. JUPITER Booster is the exascale-level module utilizing massive parallelism through GPUs for maximum performance with high energy efficiency (FLOP/J). JUPITER Cluster is a smaller general-purpose module employing state-of-the-art CPUs for applications with lower inherent parallelism and stronger memory demands. Both compute modules are procured jointly, together with a third module made of high-bandwidth flash storage. The benchmark suite has dedicated benchmarks for all modules, partly even benchmarks spanning Cluster and Booster, dubbed *MSA benchmarks*. Execution targets of the benchmarks are listed in the last columns of Table II.

During the procurement, the results of the execution of the benchmark suite for a given system proposal are weighted and combined to compute a value-for-money metric. The outcomes are compared and incorporated with other aspects into the final assessment of the system proposals.

### C. Implementation for JUPITER

The previous two sections outline general requirements for the JUPITER Benchmark Suite. Their assessment and implementation is laid out in the following and are visually sketched in Fig. 1.

Based on an analysis of current and previous compute-time allocations on predecessor systems, the suite covers a variety of scientific areas and includes applications from the domains of Weather and Climate, Neuroscience, Quantum Physics, Material Design, Biology, and others. Beyond that, diversity in workloads is realized: Artificial-Intelligence-based (AI) methods as well as classical simulations, codes based on C/C++ and Fortran, OpenACC and CUDA. Various application profiles are included, such as memory-bound or sparse computations. Future trends of workloads, e.g., the uptake of machine learning algorithms, are inferred from general trends in research communities and from recent changes in allocations on the predecessor system.

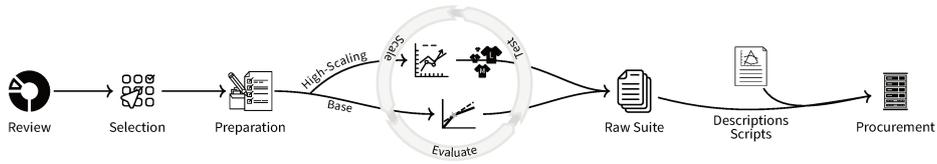


Fig. 1. Major steps in the creation of the JUPITER Benchmark Suite. Based on current workloads, a set of applications is selected. Benchmarks are prepared and then optimized in a feedback loop. Finally, descriptions are revised, and the suite is packaged for procurement.

The requirement of replicability is met by streamlining and automating benchmark execution employing the JUBE framework [3]. Reproducibility is ensured by extensive documentation of all components and the verification of computational results. Thorough testing ensures stable execution in different environments, e.g., with a varying number of compute devices, and lowers risks of commitments by vendors. Reusability is accomplished by a modular design, effective project management workflows, clear licensing, and open source publication. To ensure a high quality of the suite, the benchmarks are standardized through a well-defined setup, with consistent directory structures, uniform descriptions, and similar JUBE configurations. The infrastructure aspects are discussed further in section III.

For each of the Base benchmarks, i.e., the sixteen benchmarks used for the TCO/value-for-money calculation, a Figure-of-Merit (FOM) is identified and normalized to a time-metric. In most cases, the FOM is the runtime of either the full application or a part of it. In case the application focuses on rates, the time-metric is achieved by pre-defining the number of iterations and multiplying with the rate. Each benchmark is executed on a certain number of nodes of the preparation system (see subsection III-A) in order to create a reference execution time. The number of nodes is usually selected to be 8, but deviations are possible due to workload-inherent aspects. This is a trade-off between resource economy and agility in benchmark creation (fewer nodes are more productive), and robustness towards anticipated generational leaps in the hardware of the envisioned system (fewer nodes incorporate latent danger to severely change application profiles when the requested runtime can be achieved with one node). The time-metric, determined on the reference number of nodes, is the value to be improved upon and committed to by proposals of system designs. The number of nodes used to surpass the time-metric can be freely specified by the proposal, but is typically smaller than the reference number of nodes.

Five applications used in the Base benchmarks are capable of scaling to the full scale of the preparation system. They form the additional *High-Scaling* category and define a set of benchmarks that aim to compare the proposed system designs with the preparation system for large-scale executions. By requirement, the future system is known to achieve 1 EFLOP/s High-Performance Linpack (HPL) performance, implying a theoretical peak performance larger than that. In JUPITER’s procurement, runtimes must be committed for the High-Scaling benchmarks using a 1 EFLOP/s(th) (1 EFLOP/s the-

oretical peak performance) sub-partition. For each high-scaling application, a workload is defined to fill a 50 PFLOP/s(th) sub-partition of the preparation system (about 640 nodes) and a  $20\times$  larger sub-partition of the future system ( $20 \times 50 \text{ PFLOP/s} = 1 \text{ EFLOP/s}$ ).<sup>1</sup> The final assessment is based on the ratio of the runtime value committed for the future 1 EFLOP/s(th) sub-partition and the reference value. When the benchmark’s specific workload configuration fills up a large portion of the GPU memory on the preparation system, there is a danger that on a proposed system the scaled-up version could become limited by available memory and not showcase an accelerator’s full compute capability. This is due to the trend of growing imbalance between the advancement of compute power and memory. To give more flexibility in system design, up to four reference variants of the respective workload are prepared, taking up 25 % (tiny, T), 50 % (small, S), 75 % (medium, M), and 100 % (large, L) of the available GPU memory on the preparation system (40 GB), respectively. The system proposal may choose the variant that best exploits the available memory on the proposed accelerator after scale-up.

The seven synthetic benchmarks are selected to test individual features of the hardware components, such as compute performance, memory bandwidth, I/O throughput, and network design. Each benchmark has an individual FOM with unique rules and possibly sub-benchmarks, evaluated distinctly.

#### D. Related Work

Benchmarks have always played an important role in HPC. One objective is the assessment and comparison of technical solutions, whether at the scale of world-leading cluster systems [4], [5] or on a smaller node-level scale [6]–[8]. Benchmarks can be specific to one application topic [9]–[11] or cover multiple areas in the form of suites [12]–[15].

Foundational work focused on identifying common patterns across various applications, with the objective of classifying them based on similar computational and communication characteristics [16]. These patterns, referred to as *dwarfs*, are defined at a high level of abstraction and are intended to capture the most relevant workloads in high-performance computing. Table I categorizes the benchmarks of the JUPITER Benchmark Suite according to these dwarfs, and also gives the predominant scientific domain a benchmark represents.

<sup>1</sup>Some benchmarks have algorithmic limitations, like requirement of powers-of-two in node counts. In this case, the smaller, closest compatible number of nodes is taken (for example, 512 nodes).

TABLE I  
RELATION OF BENCHMARKS OF THE JUPITER BENCHMARK SUITE TO  
DOMAINS<sup>§</sup> AND *Berkeley Dwarfs* [16]; OTHER USE-CASES OF THE  
APPLICATIONS MIGHT HAVE OTHER PROFILES.

Benchmark	Domain	Dwarfs						
		Dense LA	Sparse LA	Spectral	Particle	Structured Grid	Unstructured Grid	Monte Carlo
Amber*	MD							
Arbor	Neurosci.	●	●				●	
Chroma-QCD	QCD		●					●
GROMACS	MD							●
ICON	Climate		●					
JUQCS	QC		●					
nekRS	CFD		●	●				
ParFlow*	Earth Sys.		●					
PIConGPU	Plasma							●
Quantum Espresso	Materials Sci.	●		●	●			
SOMA*	Polymer Sys.				●			●
MMoCLIP	AI (MM)	●						
Megatron-LM	AI (LLM)	●						
ResNet*	AI (Vision)	●						
DynQCD	QCD		●					●
NASStJA	Biology							●
Graph500	Graph							
HPCG	CG		●					
HPL	LA	●						
IOR	Filesys.							
LinkTest	Network							
OSU	Network							
STREAM	Memory							

*Graph Traversal (D. 9)*  
*Input/Output*  
*P2P, Topology*  
*Message Exchange, DMA*  
*Regular Access*

\* The benchmarks were prepared for the procurement, but not actually used.

§ The following abbreviations are used: *MD* - Molecular Dynamics; *QCD* - Quantum Chromo Dynamics; *QC* - Quantum Computing; *CFD* - Computational Fluid Dynamics; *MM* - Multi-Modal; *LLM* - Large Language Model; *LA* - Linear Algebra; *P2P* - Point-to-Point; *DMA* - Direct Memory Access.

While dwarfs can be viewed as a set of blueprints for application-inspired synthetic benchmarks, their simplicity by design prevents them to fully capture all dynamics of real applications. To address this issue in supercomputer co-design, more complex and versatile computational motifs, termed *octopodes*, are proposed by Matsuoka, Domke, Wahib, *et al.* [17].

SPEC [18], [19] is one of the most extensive, well-known benchmarking initiatives aimed at commercial users. The benchmark suite *SPECaccel2023* uses the offloading APIs OpenACC and OpenMP to measure performance with computationally intensive parallel applications, following the principle “same source code for all”. Benchmarks for the use case of system design and exascale procurement [20] have specific requirements (see subsection II-A) and are typically not made publicly available due to concerns regarding sensitive information and elaborate implementation. The PRACE Unified European Applications Benchmark Suite [21] represents a step towards a culture of open sharing, but its future support is uncertain. Other notable efforts include the CORAL-2 benchmarks [22] used for procurement of the three exascale systems in the US, and the recently developed NERSC-10 Benchmark

Suite [23] used in an ongoing procurement. HPC centers can benefit from each other’s experience, driven by a spirit of Open Science, reproducibility, and sustainable software development [1]. The integration of DevOps principles, such as Continuous Benchmarking, is gaining popularity to support these aims [24], [25].

### III. BENCHMARK INFRASTRUCTURE

#### A. Preparation Systems

The JUPITER Benchmark Suite was prepared on JUWELS, in particular JUWELS Booster, a top 20 system [26] hosted at JSC [27]. JUWELS Booster was installed in 2020 and provides a performance of 73 PFLOP/s(th) peak and 44 PFLOP/s for the HPL. The system is connected to the JUST 5 storage system [28]. JUWELS Booster provides 936 GPU nodes integrated into 39 Eviden BullSequana XH2000 racks, with 2 racks (48 nodes) building a *cell* in the DragonFly+ topology of the high-speed interconnect. Each node has 4 NVIDIA A100 GPUs and 4 NVIDIA Mellanox InfiniBand HDR200 adapters, with one adapter available per GPU. 2 AMD EPYC Rome 7402 CPUs (2 × 24 cores) are connected to 512 GB DDR4 memory.

Preparations for the High-Scaling benchmarks utilized a 50 PFLOP/s(th) sub-partition of the JUWELS Booster.

JUWELS provides general software dependencies through EasyBuild [29]. Reproducibility is achieved by either using upstream installation recipes, *easyconfigs*, or upstreaming custom recipes.

#### B. JUBE

Every benchmark is implemented in the JUBE [3] workflow environment to facilitate productive development and reproducibility. In benchmark-specific definition files, *JUBE scripts*, parameters and execution steps (compilation, computation, data processing, verification) are defined. These are then interpreted by the JUBE runtime, resolving dependencies and eventually submitting jobs for execution to the batch system. By inheriting from system-specific definition files, *platform.xml*, batch submission templates are populated and independence of the underlying system is achieved. The various sub-benchmarks and variants are implemented by *tags*, which select different versions of parameter definitions. After execution, the benchmark results are presented by JUBE in a concise tabular form, including the FOM.

Within the JUPITER procurement, the JUBE scripts are part of the documentation. They exactly define execution parameters and instructions with descriptive annotations. A textual documentation is provided as part of the benchmark-accompanying description.

#### C. Descriptions

Beyond the execution reference through JUBE scripts, each benchmark is accompanied by an extensive description. All descriptions are normalized, using identical structure with similar language. Example parts are information about the source and the compilation, execution parameters and rules,

TABLE II

OVERVIEW OF COMPONENTS OF THE JUPITER BENCHMARK SUITE. SOME DEFINING DETAILS ARE GIVEN. IF USED, SIGNIFICANT LIBRARIES ARE SHOWN (ALL BENCHMARKS USE MPI FOR DISTRIBUTION). FOR **HIGH-SCALE** BENCHMARKS, THE SUPER-SCRIPT INDICATES THE AVAILABLE MEMORY VARIANTS (TINY, SMALL, MEDIUM, LARGE). FOR **MODULE/DEVICE**, THE FOLLOWING ABBREVIATIONS ARE USED:  $B_G$  – EXECUTION ON GPUS OF JUPITER BOOSTER,  $B_C$  – EXECUTION ON THE CPUS OF JUPITER BOOSTER,  $C_C$  – EXECUTION ON THE CPUS OF JUPITER CLUSTER,  $M_{GC}$  – MSA EXECUTION WITH CPUS OF JUPITER CLUSTER AND GPUS OF JUPITER BOOSTER.

	Application Features			Execution Targets		
	Benchmark Name	Progr. Language, [Libraries, ]Prog. Models	Licence	Nodes Base	Nodes High-Scale $N^{Mem\ Vars}$	Module/Device $B_G B_C C_C M_{GC}$
Application	Amber*	Fortran, CUDA	Custom	1		✓
	Arbor	C++, CUDA/HIP	BSD-3-Clause	8	642 <sup>T,S,M,L</sup>	✓
	Chroma-QCD	C++, QUDA, CUDA/HIP	JLab	8	512 <sup>S,M,L</sup>	✓
	GROMACS	C++, CUDA/SYCL	LGPLv2.1	3/128		✓
	ICON	Fortran/C, OpenACC/CUDA/HIP	BSD-3-Clause	120/300		✓
	JUQCS	Fortran, CUDA/OpenMP	None	8	512 <sup>S,L</sup>	✓
	nekRS	C++/C, OCCA, CUDA/HIP/SYCL	BSD-3-Clause	8	642 <sup>S,M,L</sup>	✓
	ParFlow*	C, Hypr, CUDA/HIP	LGPL	4		✓
	PICongPU	C++, Alpaka, CUDA/HIP	GPLv3+	8	640 <sup>S,M,L</sup>	✓
	Quantum Espresso	Fortran, ELPA, OpenACC/CUF	GPL	8		✓
	SOMA*	C, OpenACC	LGPL	8		✓
	MMoCLIP	Python, PyTorch, CUDA/ROCM <sup>1</sup>	MIT	8		✓
	Megatron-LM	Python, PyTorch/Apex, CUDA/ROCM <sup>1</sup>	BSD-3-Clause	96		✓
	ResNet*	Python, TensorFlow, CUDA/ROCM <sup>1</sup>	Apache-2.0	10		✓
	DynQCD	C, OpenMP	None	8		✓
	NASJA	C++, MPI	MPL-2.0	8		✓
	Synthetic	Graph500	C, MPI	MIT	4/16/all	
HPCG		C++, OpenMP, CUDA/HIP	BSD-3-Clause	1/4/all		✓
HPL		C, BLAS, OpenMP, CUDA/HIP	BSD-4-Clause	1/16/all		✓
IOR		C, MPI	GPLv2	-/> 64 <sup>§</sup>		✓
LinkTest		C++, MPI/SIONlib	BSD-4-Clause+	all		✓
OSU		C, MPI, CUDA	BSD	1/2		✓
STREAM		C, CUDA/ROCM/OpenACC	Custom	1		✓
					$N^{Mem\ Vars}$	$B_G B_C C_C M_{GC}$

\* The benchmarks were prepared for the procurement, but not actually used.

<sup>1</sup> For PyTorch and TensorFlow, CUDA and ROCm backends are available; through *extensions*, also backends for Intel GPUs exist (not in mainline repositories).

<sup>§</sup> IOR features two sub-benchmarks, *easy* and *hard*. The number of nodes is a free parameter in *easy*. In *hard*, it can also be chosen freely, as long as more than 64 nodes are taken.

detailed instructions for execution and verification, sample results, and concluding commitment requests. In all relevant sections, relations to the JUBE scripts are made in addition to the textual descriptions. For the vendors, the use of JUBE is recommended but not mandatory.

PDFs generated from the benchmark descriptions are part of the committed procurement documentation, including hashes of archived benchmark repositories.

#### D. Git and Submodules

All components of a benchmark are available in a single Git repository as a single source of truth. A common structure is established, containing description, JUBE scripts, auxiliary scripts, benchmark results, and the source code of the benchmarked application. Utilizing the attached issue tracker, project management and collaboration are facilitated.

Per default, the sources are included as references in the form of Git Submodules. Submodules enable a direct linkage

to well-defined versions of source code, but do not unnecessarily clutter the benchmark repository by static, potentially extensive copies. They are well integrated into the Git workflow and easily updated. In cases where inclusion as a Git Submodule is not possible, scripts and detailed instructions for download are provided.

For delivery as part of the procurement specification package, each benchmark repository is archived as a tar file.

If too large for inclusion in the Git repository, input data is provided as a separate download, including a verifying hash.

#### E. Project Management

The benchmark suite development efforts were supported efficiently by clear project management workflows over several months.

A core team of HPC specialists and scientific researchers initiated the process early, curating a list of potential benchmarks based on their expertise and experience with existing

HPC systems, while also incorporating insights from previous procurements. In close collaboration with domain scientists, this list was gradually refined to ensure a balanced and diverse selection that met the requirements outlined in subsection II-A.

Competent teams, each led by a team captain, were responsible for individual applications. GitLab issues were used to document biweekly meetings and track per-application progress in the form of a pre-defined checklist with 11 points (ranging from source code availability, over JUBE integration, to description creation). Collaborative hack days facilitated collaboration while running the benchmarks on the preparation system.

#### IV. BENCHMARKS

This section describes the JUPITER Benchmark Suite, containing 23 benchmarks: 7 synthetic and 16 application benchmarks. In the procurement process, the number of application benchmarks was reduced to 12. Table II gives an overview of all benchmarks, including application features and execution targets.

With this work, the JUPITER Benchmark Suite is released as open source software at <https://github.com/FZJ-JSC/jubench>, with individual repositories for each benchmark [30]–[52].

##### A. Application Benchmarks

In the following, we describe in detail eleven of the application benchmarks, divided into Base and High-Scaling benchmarks. The benchmarks are based on prominent workloads in the HPC community and were specifically developed for the suite. Given their complex computational dynamics, a certain level of technical and scientific background is necessary and will be provided accordingly. The remaining benchmarks, Amber, ParFlow, SOMA, ResNet, and DynQCD, are briefly introduced first for completeness; they are either using closed-source software (DynQCD) or were ultimately not used for the JUPITER procurement (the others).

- *Amber* [53], [54] is a popular commercial molecular dynamics code for biomolecules. The Satellite Tobacco Mosaic Virus (STMV) case from the Amber20 benchmark suite [55] (1 067 095 atoms) is chosen. The code is mainly optimized for single GPU calculations and is not intended to scale beyond a single node.
- *ParFlow* [56], [57] is a massively-parallel, open source, integrated hydrology model for surface and subsurface flow simulation. The ClayL test from ParFlow’s test suite (simulating infiltration into clay soil) is selected, with a problem size of  $1008 \times 1008 \times 240$  cells [58].
- *SOMA* [59] performs Monte Carlo simulations for the “Single Chain in Mean Field” model [60], studying the behaviour of soft coarse-grained polymer chains in a solution.
- *ResNet* [61] uses convolutions and residual connections for training deep neural networks and serves as a reference model in computer vision tasks. The suite includes ResNet50, implemented in TensorFlow with Horovod.

- *DynQCD* [62] is a CPU-only code which performs numerical simulations for Lattice Quantum-Chromodynamics (LQCD). The benchmark generates 600 quark propagators using a conjugate gradient solver for sparse LQCD fermion matrices, with high demands to the memory sub-system.

1) *Base Benchmarks (Selection)*: The Base benchmarks are designed to incentivize system designs that optimize the time to solution. They are first executed on a reference number of nodes on the preparation system (see subsection II-C). Figure 2 gives an overview of application runtimes and respective strong scaling behaviors for surrounding number of nodes. While the absolute number can be used to judge system designs quantitatively, the strong scaling behavior can be used as an additional data point to understand the overall design qualitatively. Note the example for reading the graph in the figure caption.

a) *GROMACS*: GROMACS [63]–[66] is a versatile package to perform molecular dynamics simulations, focusing on biochemical molecules and soft condensed matter systems. The application integrates Newton’s equations of motion for systems with hundreds to millions of particles and provides time-resolved trajectories. Two biological systems from the Unified European Applications Benchmark Suite (UEABS) [21] are used, test cases A and C. Test case A simulates a GluCl ion channel embedded in a membrane. Test case C contains 27 replicas of the STMV with about 28 000 000 atoms and allows testing the scalability of system-supplied Fast Fourier Transform (FFT) libraries.

b) *ICON*: The ICOSahedral Non-hydrostatic model (ICON) [67] is a modelling framework for weather, climate, and environmental prediction used for operational weather forecasting at the German Weather Service. ICON also provides an Earth System Model for climate simulations, i.e., a general circulation model of the atmosphere, including a land module [68] and an ocean model [69]. While the atmosphere part has been ported to GPUs [70], the ocean component is still running on CPUs only. ICON is available under a permissive open source licence. The JUPITER benchmark case is based on the atmosphere component, with global forecast simulation in two resolutions, resulting in two sub-benchmarks: R02B09 (5 km grid point distance) and R02B10 (2.5 km grid point distance) [71]. The coarser resolution is targeted for execution on 120 nodes, and the finer resolution is for 300 nodes. While reasonable scaling to  $2 \times$  the node count (240 nodes and 600 nodes, respectively) is possible, this is not the usual mode of operation for ICON. These simulations are crucial for ICON’s development towards a storm-resolving climate model with 1 km resolution or even less [72]. A unique aspect of the ICON benchmark is its large input dataset: R02B09 requires 1.8 TB of data, R02B10 needs 4.5 TB. Therefore, the ICON benchmark also tests the performance of I/O operations on a system.

c) *Megatron-LM*: Megatron-LM [73] is a prominent codebase in Natural Language Processing (NLP), known for its vast scale and performance capabilities. It employs the

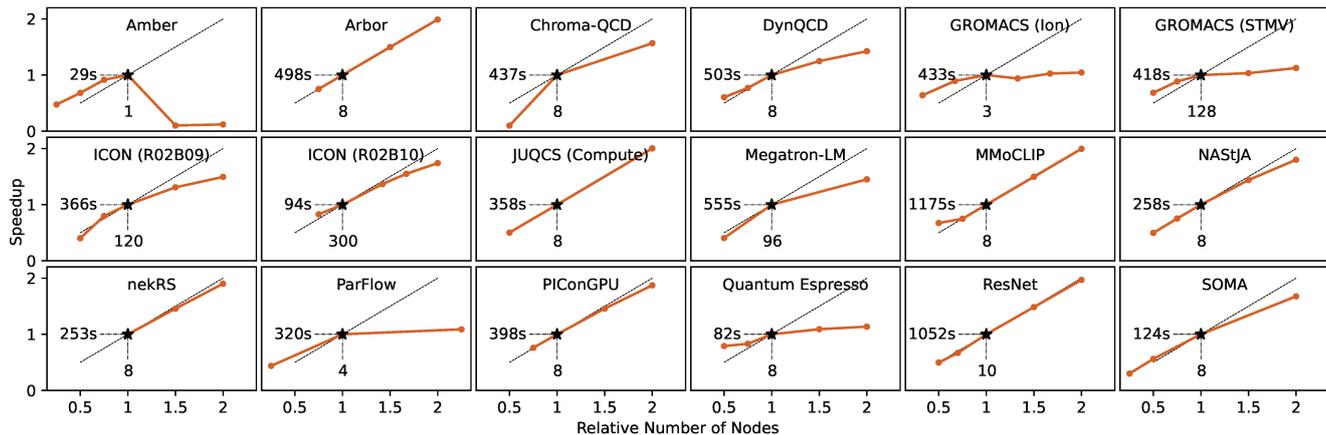


Fig. 2. Overview of relative runtimes of all Base applications on the reference system, JUWELS Booster. Shown at (1, 1) is the execution on the reference number of nodes with the reference runtime, with the absolute values shown close to the horizontal and vertical axes, respectively. Beyond the reference execution, strong-scaled relative runtimes (with respect to the reference runtime) on the surrounding number of nodes are given (usually  $0.5\times$ ,  $0.75\times$ ,  $1.5\times$ , and  $2\times$  the reference; some benchmarks deviate). As an example, consider Arbor: the reference number of nodes (8) is noted at horizontal 1, the reference runtime of 498 s at vertical 1; further data-points are given for 4 nodes (663 s), 12 nodes (332 s), and 16 nodes (250 s) –  $0.5\times$ ,  $1.5\times$ , and  $2\times$  the reference number of nodes of 8. See also Table II.

transformer model architecture [74] and leverages various parallelization techniques and optimizations [75]–[78] through PyTorch to achieve high hardware utilization with excellent efficiency. The training of various recent open source GPT-like models was carried out with Megatron-LM [79], making this benchmark crucial to assess a system’s capability to handle disruptive generative AI workloads. The benchmark trains a 175 billion parameter model, converting the usual throughput metric (tokens per time) to a hypothetical time-to-solution FOM by training 20 million tokens.

*d) MMoCLIP:* Contrastive Language-Image Pre-training (CLIP) [80] is a method that conducts self-supervised learning on weakly-aligned image-text pairs with open vocabulary language, resulting in language-vision foundation models. The approach enables usage of substantially increased datasets, like web-scale datasets [81], [82]. The OpenCLIP [83], [84] codebase, an open source implementation of CLIP, enables efficiently distributed training of CLIP models by using multiple data parallelism schemes through PyTorch, scaling to more than a thousand GPUs. Due to its strong transferability and robustness, OpenCLIP is used in diverse multi-modal learning approaches and downstream applications [85]–[87], and efficient training is crucial for the machine learning community dealing with open, fully reproducible foundation models.

The MMoCLIP benchmark is curated from OpenCLIP. It trains an ViT-L-14 model on a synthetic dataset of 3 200 000 image-text pairs and records the total training time as a FOM.

*e) Quantum ESPRESSO:* Quantum ESPRESSO (QE) [88]–[90] is an open source, density-functional-theory-based electronic structure software used both in academia and industry. QE calculates different material properties using a plane wave basis set and pseudo-potentials and can exploit novel accelerators well [91]. The dominant kernel in

QE performs a three-dimensional FFT, which is usually a memory-bound kernel and is communication-bound for large systems [91].

For the benchmark suite, the *Car-Parrinello Molecular Dynamics* model was chosen. The benchmark is based on a use case created in the MaX project [92], [93] and does calculations for a slab of  $\text{ZrO}_2$  with 792 atoms.

*f) NASTJA:* The Neoteric Autonomous Stencil code for Jolly Algorithms (NASTJA) is a massively-parallel simulation framework of biological tissues using a Cellular Potts Model [94], [95]. This model relies on nearest neighbour interactions and is parallelized by dividing the overall workload into multiple sub-regions, called blocks. Each block is treated independently by an MPI process, with boundaries being exchanged. Using NASTJA, tissues composed of thousands to millions of cells can be simulated at subcellular resolution [96]. As a test case, adhesion-driven cell sorting is used, a common process in tissue development and segregation [97]. The benchmark investigates the first 5050 Monte Carlo (MC) steps of a system of size  $720 \times 720 \times 1152 \mu\text{m}^3$ , containing roughly 600 000 cells. NASTJA utilizes MPI for parallelization/distribution and is one of the few CPU-only benchmarks in the suite. The application exhibits an irregular memory access pattern at each iteration, which is not suitable for GPU execution.

*2) High-Scaling Benchmarks:* In this section, we describe the five High-Scaling benchmarks in detail.

Figure 3 juxtaposes the weak-scaling behaviours of the applications over a wide range of node numbers, using the reference HPC system JUWELS Booster.

*a) Arbor:* Arbor is a library for simulating biophysically-realistic neural networks, bridging the gap between point and nanoscale models [98]. Developed in the HBP [99], it aims at efficient use of modern HPC hardware behind an

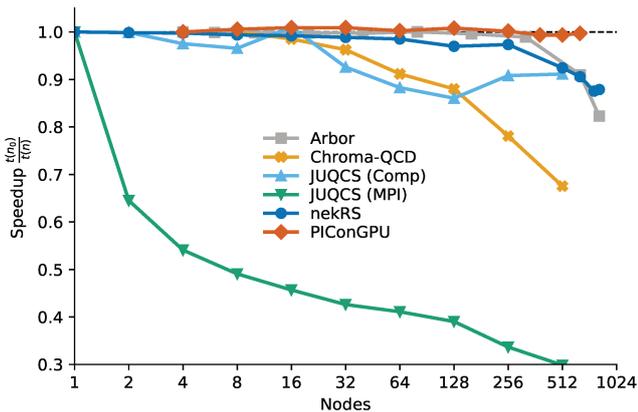


Fig. 3. Weak scaling efficiency of the five High-Scaling benchmarks over a wide range of JUWELS Booster node numbers. For JUQCS, two lines are drawn; one for the computation and one for the communication (see section IV-A2c).

intuitive interface. Neurons are modelled by morphology, ion channels, and connections. Arbor is written in C++ with a Python interface and available under a permissive open source license. The user-centric description is discretized and aggregated to optimize data layouts for individual hardware. At runtime, the *cable equation* is integrated alternating with a system of ODEs for the channels. Users model channels via a domain-specific language that must be compiled for the target hardware. Communication is performed, concurrently with time evolution, every  $n$  steps, determined by neural delays. The benchmark is parameterized to fill the GPU memory in the variants T, S, M, L. To differentiate from point models, it is weighted heavily towards computation, emphasized by a sparse connectivity. A complex cell from the Allen Institute was selected and adapted to random morphologies of fixed depth [100]. Cells are organized into rings propagating a single spike. Rings are interconnected to place load on the network without altering dynamics, yielding a deterministic, scalable workload. Profiling shows two cost centers: 52% ion channels and 33% cable equation; hiding communication completely. The Base version runs on 32 JUWELS Booster nodes, filling all 4 GPUs’ memory. This was scaled up to the full Booster to verify efficient resource usage and extrapolated to 1 EFLOP. The number of generated spikes is used for validation.

b) *Chroma*: Chroma [101] is an all-purpose application for LQCD computations. It is compiled on top of the USQCD software stack [102], which provides LQCD-specific libraries for communication, data-parallelism, I/O, and, importantly, sparse linear solver libraries optimized for different architectures. Key libraries used in this benchmark include QMP for MPI wrapping, QDP-JIT for data-parallelism and parallel-I/O via QIO, and the GPU-targeted QUDA solver library [103]. Chroma and the USQCD stack are open source and community-developed. Chroma is one of the most widely used LQCD suites and is representative of LQCD codes in general.

LQCD calculations generally depend heavily on solving very large, regular, sparse linear systems (dimension  $10^6 - 10^9$  generally). Due to the regularity of the data and the calculation, LQCD lends itself to many levels of concurrency.

The Chroma LQCD benchmark in the JUPITER Benchmark Suite contains the representative benchmarks for the Hybrid Monte-Carlo (HMC) component of the LQCD simulations. In the benchmark, a number of HMC update trajectories are performed using the  $3 + 1$  flavours of Clover Wilson fermions — three light quark flavours with identical mass, and a fourth flavour with heavier mass — and the Lüscher-Weisz gauge action. The 4D lattice is initialized with a random  $SU(3)$  element on each link. Checkpointing is disabled by a source patch to remove the I/O overhead for the calculations. The benchmark also contains a fix to Chroma, allowing simulation of 4D lattice volumes greater than  $2^{31}$  and among the largest LQCD simulations anywhere to date. The benchmark performance is sensitive to the decomposition configuration used for distributing the 4D lattice to different tasks and to the affinity between the CPU, GPU, NUMA domains, and the network controller for each task.

The benchmark is validated by comparing the output with a reference solution with a tolerance of  $10^{-10}$  for the Base benchmark and  $10^{-8}$  for High-Scaling benchmarks.

The relevant metric (FOM) is the total time spent in HMC updates, excluding the first update, which includes overhead for tuning QUDA parameters. So a minimum of two updates must be prescribed.

c) *JUQCS*: JUQCS is a massively parallel simulator for universal gate-based Quantum Computers (QCs) [104] written in Fortran 90 using MPI, OpenMP, and CUDA. During the past decades, JUQCS has been used to benchmark some of the largest supercomputers worldwide, including the Sunway TaihuLight and the K computer [105] as well as JUWELS Booster [106], and it was part of Google’s quantum supremacy demonstration [107]. Various versions of JUQCS are available in binary form as part of a container [108]; a light version with available sources was created specifically for this benchmark suite [109]. JUQCS simulates an  $n$ -qubit gate-based QC by iteratively updating a rank- $n$  tensor of  $2^n$  complex numbers (state vector) stored in double precision and distributed over the supercomputer’s memory. The total available memory determines the size of the largest QC that can be simulated. For instance, a universal simulation of  $n = 45$  qubits requires a little over  $16 \times 2^{45} \text{ B} = 0.5 \text{ PiB}$ . Many operations require the transfer of half of all memory, i.e.,  $2^n/2$  complex double-precision numbers, across the network, which can help to assess the performance of a supercomputer’s communication network [105], [106]. As Fig. 3 shows, the deviation of JUQCS w.r.t. the theoretically expected linear scaling (green triangles) reveals both a drop in performance from intra-node to inter-node GPU communication (from 1 to 2 nodes) and another drop when communication enters the large-scale regime at 256 nodes.

All present JUQCS benchmarks simulate successive applications of a single-qubit quantum gate that requires large

memory transfers. The Base benchmark simulates  $n = 36$  qubits requiring 1 TiB of GPU memory. The High-Scaling benchmark contains two memory variants: a large memory variant with  $n = 42$  qubits requiring 64 TiB (L) and a small memory variant with  $n = 41$  qubits requiring 32 TiB of GPU memory (S). Rules are given for extrapolation to an Exascale setup using  $n = 46$  (L) or  $n = 45$  qubits (S). For all test cases, verification is done using the theoretically known results [105].

In addition, an MSA version of the JUQCS benchmark simulates  $n = 34$  qubits on both JUWELS Cluster and Booster simultaneously. The total amount of memory is split into two parts, with 128 GiB residing on the CPU nodes and 128 GiB residing on the GPU nodes. MPI is used for communication between the Cluster and the Booster, and the number of MPI tasks is similarly split into two. On the Cluster, each MPI task launches 12 OpenMP threads, with one thread per CPU core. On the Booster, each MPI task controls one of the GPUs.

*d) nekRS:* nekRS [110] is a fast CFD solver designed for GPUs that solves the low-Mach Navier-Stokes equations (NSEs), potentially coupled with multiphysics effects. nekRS has been run at scale on many large supercomputers, featuring excellent time-to-solution due to its high GPU throughput, and was nominated for the 2023 Gordon-Bell Award [111]. nekRS uses high-order spectral elements [112] in which the solution, data, and test functions are represented as locally structured  $N^{\text{th}}$ -order tensor product polynomials on a set of  $E$  globally unstructured curvilinear hexahedral brick elements. There are two key benefits to this strategy. First, high-order polynomial expansions significantly reduce the number of unknowns ( $n \approx EN^3$ ) needed to achieve engineering tolerances. Second, the locally structured forms allow tensor product sum factorization, which yields low  $\mathcal{O}(n)$  storage cost and  $\mathcal{O}(nN)$  work complexity [113]. The leading order  $\mathcal{O}(nN)$  work terms can be cast as small dense matrix-matrix products with good computational intensity [114]. nekRS is written in C++ and the kernels are implemented using the portable Open Concurrent Compute Abstraction (OCCA) library [115] for abstraction between different parallel languages/hardware architectures.

The benchmark case is derived from a Rayleigh-Bénard convection (RBC) application [116], [117] which simulates turbulence induced by a temperature gradient — a typical case executed at scale. The simulation domain is a *sheet*. It is much more extended in the periodic directions than in the wall-bounded direction. The chosen polynomial order is 9 with 600 time steps per run. Verification is based on pre-computed results and derived tolerances. The High-Scaling benchmark variants use between 28 836 900 (*small*,  $\sim 11\,229$  per GPU) and 57 760 000 (*large*,  $\sim 22\,492$  per GPU) elements, which is more than the minimum number of elements required for the "strong scaling limit" of 7000 – 8000 elements per GPU. The Base benchmark case uses 719 104 elements resulting in 22 472 elements per GPU.

*e) PIConGPU:* PIConGPU is an open source, fully relativistic particle-in-cell (PIC) code designed for studying laser-plasma interactions and astrophysical phenomena. It uses the PIC algorithm with several key components, namely, parti-

cle initialization, charge calculations using grid interpolation, field calculations using densities, and time-marching due to Lorentz force. This approach allows particles to interact via fields on the grid rather than direct pairwise interactions, reducing computational steps from  $N^2$  to  $N$  for  $N$  particles. PIConGPU employs a unique data model with asynchronous data transfers to handle the computational challenge. It can simulate complex plasma systems with billions of particles on GPU clusters [118]. PIConGPU is developed with a hardware-agnostic approach using the Alpaka library [119], [120], providing outstanding performance across all supported platforms, like CPUs, AMD and NVIDIA GPUs, and FPGAs [121]. The benchmark suite uses a 3D test-case simulating the Kelvin-Helmholtz Instability (KHI), a non-relativistic shear-flow instability, utilizing a pre-ionized hydrogen plasma with periodic boundary conditions. While relevant for various research communities, the nature of shear-flows and the use of periodic boundary conditions does not impose a significant load imbalance throughout the simulation. Therefore, the performance of the code is based on its structure rather than the physics of the problem. In the KHI use case, the number of particles per cell is kept constant to 25, using as many cells as the GPU memory allows. A grid size of  $\vec{x} = (4096, 2048, 1024)$  is chosen for the small memory variant, and extended to  $(4096, 2048, 2048)$  (M) and  $(4096, 4096, 2560)$  (L) for the larger variants. PIConGPU employs domain decomposition for distribution, dividing the computational domain into smaller subdomains along three dimensions. To distribute along these three dimensions, the maximum number of nodes that can be utilized is limited to 640, rather than 642.

## B. Synthetic Benchmarks

The JUPITER Benchmark Suite also includes seven well-known synthetic benchmarks: Graph500, HPCG, HPL, IOR, LinkTest, OSU, and STREAM (including a GPU variant). The IOR and LinkTest implementation are presented in the following, highlighting some unique aspects of the setup.

*a) IOR:* IOR is the de facto standard for measuring I/O performance and is being used by the IO500 [122] to compare the I/O characteristics of storage systems. The benchmark provides a large list of parameters such as block size, transfer size, API, and task reordering, which in turn allows simulating multiple I/O patterns. To target the high-bandwidth, NVM-based JUPITER storage module, the upper and lower bounds on the mean read and write bandwidth are our focus in the Benchmark Suite. Similar to IO500, two variants of the IOR are implemented, *Easy* and *Hard*. The Easy variant requires a transfer size of 16 MiB, with each process writing to its own file. The Hard variant uses a transfer size of 4 KiB and a block size of 4 KiB, with all processes writing and reading a single file. The setup forces multiple processes to write to the same file system data block, stressing the filesystem with the lock processes. The remaining parameters were selected to avoid caching effects. The number of nodes is a free parameter (with a lower bound) to allow for optimization on the level of parallelism that the underlying file system can provide.

b) *LinkTest*: LinkTest [123] is designed to test point-to-point connections between processes in serial or parallel mode and is capable of handling very large numbers of processes. It is an essential tool in network operations, used mostly internally by system administrators for acceptance testing, maintenance, and troubleshooting.

The JUPITER Benchmark Suite utilizes LinkTest’s *bisection* test, to concisely evaluate interconnectivity between parts of the system’s network, quantified by a single metric. In the bisection test, a number of test processes (one per high-speed network adapter) is separated to two equal halves of the system, and messages are bounced between partnering processes in parallel (bidirectional mode). To achieve optimal bandwidth, the message size is set to 16 MiB. An assessment is made mainly based on the minimum bisection bandwidth.

## V. LESSONS LEARNED

We developed the JUPITER Benchmark Suite building upon our experience from previous HPC system procurements. The suite constitutes a substantial expansion from those earlier endeavours, and should be considered as a living object that will continue to evolve over time. In the following, we summarize the lessons learned, covering first the perspective of application developers, then the benchmark suite creation, and finally the overall procurement process.

### A. Application Development

The applications of the JUPITER Benchmark Suite not only need to be executed on current large-scale resources like JUWELS Booster, but also need to be extrapolated to larger future resources, amplifying scaling challenges.

To understand the performance characteristics on a future system better, it proved useful for some application developers to create **models** of their applications. JUQCS, for example, has a non-trivial weak scaling behaviour. In the benchmark, the execution time is reported in relation to an ideal scenario, enabling comparability. A model was developed for nekRS to predict the performance of a later part of the simulation early in the process, allowing much shorter and more resource-efficient benchmarks. During scalability studies for PICongPU, a model for the scaling behaviour could be developed, informing valid simulation parameters for the benchmark setup.

While the approximate scale of the future system is known, the details of the setup are not. When domain decomposition is important for performance, preliminary studies are usually employed to determine the best parameters for production runs. But **decomposition studies** are impossible in the benchmark context, especially for an unknown system design. Through labour- and resource-intensive investigation, estimates, rules, or scripts for ideal domain decomposition were devised, e.g., for Chroma-QCD, PICongGPU, NASTJA, and DynQCD. This also documented the experience of individual researchers, improving reproducibility. To understand different scaling regimes of the application, a network communication model was developed for JUQCS. The model can be employed

to understand topological aspects of the high-speed network of current and future systems, for example with respect to congestion.

The preparation for **future system designs** had a direct effect on application development. For example, it became apparent for Arbor developers that they need to optimize memory usage, as memory capacity and bandwidth will continue to extend more slowly compared to compute performance. During benchmark preparation, they also needed to trade highly-valued user experience for scalability, as the approach of referring to connection endpoints with labels did not scale as required. A short-term solution (using local indexing) was found for the suite, and a hash-based solution is being developed upstream. On a similar note, Chroma-QCD authors needed to patch the code to facilitate execution on the envisioned scale. Unexpected effects can occur depending on the extrapolation method to the future system. For Chroma-QCD, it was found out that the employed benchmark is not guaranteed to converge, and a cut-off after a certain number of iterations is a more robust approach.

Result **verification** is essential for a benchmark suite to ensure the validity of submitted results. Yet, the experience with verification during the suite preparation varied. Some results could be verified either exactly (JUQCS), or within a certain numerical limit by comparing to a pre-computed solution (Chroma-QCD); more involved simulations were verified by extracting key metrics from the computed solution for comparison to a model (ICON, nekRS). The verification of some applications with iterative algorithms, which were stopped before convergence, relied on framework-inherent verification and required key data in the output (PICongGPU, Megatron-LM) — arguably the weakest form of verification.

### B. Benchmark Design

Creating a vast benchmark suite that picks up the status quo in workloads, and bringing it to mature levels, is a **resource-intensive endeavour**. Beyond the human resources, compute resources of the reference system constitute a significant investment. To use these resources efficiently, it is important to design benchmarks with runtimes as short as possible, while keeping it as large as needed. Short runtimes also enable swift turn-around times for rapid prototyping – especially useful in a large suite. The size of the input data and the files generated at runtime should be minimized to ensure that the benchmark suite is easy to handle. For reproducibility, non-core application parts like pre-processing or data-staging should be kept short. Modelling domain decomposition effects, also beyond typical production execution profiles, is further consuming resources.

Preparing for target system designs with **unknown details and scales** beyond available resources is a demanding task. Care should be taken to consider future hardware trends in the benchmark design. To explicitly accommodate different compute-to-memory ratios, up to four memory variants of benchmarks were introduced. On the preparation system, the memory variants can be used to study artificially-limited com-

pute profiles and determine possible bottleneck shifts on future systems. With unknown hardware, algorithmic behaviours might shift as well, and iterations may not converge. A more robust approach is to not compute until convergence, but stop after a predetermined amount of iterations.

Some parameters in the benchmark suite are free to choose, like the number of nodes, others are fixed, for example simulation parameters. Thorough **execution rules** and modification guidelines determine the envisioned outcome and need to be developed as part of the suite. Beyond general rules, benchmarks may explicitly deviate to either loosen or tighten rules. Parameter validation should be part of the overall verification process; further extending on the importance of the verification task.

### C. Procurement

Using application benchmarks in the procurement of HPC systems is essential to realistically represent user requirements when deciding the configuration of the future system. An additional challenge was given by the particular system architecture of JUPITER, in which **two compute modules** of very disparate sizes, a small CPU-only module and an exascale GPU-accelerated module, are coupled together with a shared storage module. It took some discussions until finding the right number and balance between CPU and GPU benchmarks, which ended up being in the ratio of about 1:5.

To formulate and develop the benchmarks, it proved fruitful to **collaborate closely** with the domain researchers intending to utilize the system. Established relationships in joint projects are especially productive, while it is more difficult for new user domains. A fundamental limitation of our approach is the **reliance on existing application codes** executed on current systems. By design, disruptive approaches are not well covered and a tendency to favour evolutionary technology is introduced. However, considering the effort associated with adopting new technologies among HPC users, this focus on incremental developments is justified. Still, predicting future system **usage trends** is crucial — like the AI applications in the suite, which aim at representing a user domain expected to gain importance over time. However, the rapidly evolving software and algorithms in this domain make it hard to accurately estimate their future needs. It is therefore important to consider the most recent breakthroughs in AI beyond the HPC context, including also commercially-driven domains.

The **time window** for the development of the JUPITER Benchmark Suite was limited by the constraints of the procurement process. The endeavour started several months before the procurement, and required dedicated work by tens of people. Clear management structures and collaboration platforms were essential **tools for extensive collaboration**. In particular, transparent communication with all bidders was crucial, which was possible thanks to the dialogue phase that was part of the procurement. The suite development fostered collaboration, team-building, and knowledge-sharing. Code and environment optimizations were openly shared between benchmark developers and vendors, iteratively improving the

benchmarks further. The suite itself is now open source and can **benefit the wider HPC community**.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper, we presented the JUPITER Benchmark Suite, which has been successfully employed in the procurement process of the exascale system JUPITER. The suite has served as a valuable tool in assessing HPC system performance during the procurement and beyond. The benchmark applications (see section IV) were chosen to represent the workloads on the future system after careful consideration of requirements and constraints. Bidders used this suite to test different technologies, put together their proposals, and prepare and commit the associated performance numbers. The procuring entity selected the vendor based on these values, together with multiple additional evaluation criteria. At the time of writing, the JUPITER system is being installed. The benchmark suite will be employed again during the acceptance procedure.

The JUPITER Benchmark Suite lays a foundation to further extend and automate HPC system benchmarking. Facilitated by the reusable design of the suite, Continuous Benchmarking will be realized as future work, employing the CI/CD features of GitLab in conjunction with novel tools such as Jacamar [25]. Running the suite at regular intervals (e.g., after maintenances), we will ensure that the system does not see performance degradation over its lifetime or after updates. Application optimization for JUPITER will continue during the system deployment and installation phase, utilizing experiences gained and tools created. We will strive for further improvements regarding the reproducibility of individual benchmarks, including a focus on verification. Also, individual technical enhancements are in progress (for example, using git-annex for the large input data).

### ACKNOWLEDGMENT

Many benchmarks presented in this work have roots in projects that received funding from national and European sources; for example, Quantum Espresso (from the MaX project [124]) or nekRS (from the CoEC project [125]). The authors thank the funding agencies for their support and the colleagues from the projects for being available for advice when designing the suite.

The authors would like to thank the following people for their contributions: Max Holicki and Yannik Müller, for their contributions to the LinkTest benchmark; Jonathan Windgassen and Christian Witzler for their support with the nekRS benchmark; Hans De Raedt for the discussions and help with the JUQCS benchmark.

The double-anonymous review of this publication was enabled by the *Anonymous GitHub* service at <https://anonymous.4open.science/>. The authors thank the creator, also for the immediate, last-minute support.

Finally, the authors would like to thank EuroHPC JU, BMBF, MKW-NRW, and GCS for the supportive atmosphere, productive discussions, and for the financial support of the JUPITER project.

## REFERENCES

- [1] J. Fehr, J. Heiland, C. Himpe, and J. Saak, “Best practices for replicability, reproducibility and reusability of computer-based experiments exemplified by model reduction software,” *AIMS Mathematics*, vol. 1, no. 3, pp. 261–281, 2016, ISSN: 2473-6988. DOI: 10.3934/Math.2016.3.261.
- [2] E. Suarez, N. Eicker, T. Moschny, *et al.*, “Modular Supercomputing Architecture: A success story of European R&D,” Tech. Rep. 9, 2022, p. 24. DOI: 10.5281/ZENODO.6508394.
- [3] T. Breuer, J. Wellmann, F. Souza Mendes Guimarães, C. Himmels, and S. Luehrs, *JUBE*, version REL-2.6.1, Nov. 2023. DOI: 10.5281/zenodo.10228432.
- [4] *The TOP500 list*, <http://www.top500.org>, Accessed: 2024-03-13.
- [5] J. J. Dongarra, P. Luszczek, and A. Petitet, “The LINPACK benchmark: Past, present and future,” *Concurrency and Computation: Practice and Experience*, vol. 15, no. 9, pp. 803–820, 2003. DOI: <https://doi.org/10.1002/cpe.728>.
- [6] A. Sorokin, S. Malkovsky, G. Tsoy, A. Zatsarinnyy, and K. Volovich, “Comparative performance evaluation of modern heterogeneous high-performance computing systems CPUs,” *Electronics*, vol. 9, no. 6, 2020, ISSN: 2079-9292. DOI: 10.3390/electronics9061035.
- [7] R. Xu, X. Tian, S. Chandrasekaran, Y. Yan, and B. Chapman, “NAS parallel benchmarks for GPGPUs using a directive-based programming model,” in *Languages and Compilers for Parallel Computing*, J. Brodman and P. Tu, Eds., Cham: Springer International Publishing, 2015, pp. 67–81, ISBN: 978-3-319-17473-0. DOI: 10.1007/978-3-319-17473-0\_5.
- [8] C. Bienia, S. Kumar, J. P. Singh, and K. Li, “The PARSEC benchmark suite: Characterization and architectural implications,” in *Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques*, ser. PACT ’08, Toronto, Ontario, Canada: Association for Computing Machinery, 2008, pp. 72–81, ISBN: 9781605582825. DOI: 10.1145/1454115.1454128.
- [9] A. Herten, T. Hater, W. Klijn, and D. Pleiter, “Performance comparison for neuroscience application benchmarks,” in *High Performance Computing*, M. Weiland, G. Juckeland, S. Alam, and H. Jagode, Eds., Cham: Springer International Publishing, 2019, pp. 418–431, ISBN: 978-3-030-34356-9. DOI: 10.1007/978-3-030-34356-9\_31.
- [10] J. Albers, J. Pronold, A. C. Kurth, *et al.*, “A modular workflow for performance benchmarking of neuronal network simulations,” *Frontiers in Neuroinformatics*, vol. 16, 2022, ISSN: 1662-5196. DOI: 10.3389/fninf.2022.837549.
- [11] S. Farrell, M. Emani, J. Balma, *et al.*, “MLPerf™ HPC: A holistic benchmark suite for scientific machine learning on HPC systems,” in *2021 IEEE/ACM Workshop on Machine Learning in High Performance Computing Environments (MLHPC)*, Los Alamitos, CA, USA: IEEE Computer Society, Nov. 2021, pp. 33–45. DOI: 10.1109/MLHPC54614.2021.00009.
- [12] J. Dongarra and P. Luszczek, “HPC challenge: Design, history, and implementation highlights,” in *Contemporary High Performance Computing: From Petascale toward Exascale*, J. S. Vetter, Ed. Chapman and Hall/CRC, 2013, vol. 1, ch. 2, ISBN: 9781466568358.
- [13] J. A. Stratton, C. I. Rodrigues, I.-J. Sung, *et al.*, “Parboil: A revised benchmark suite for scientific and commercial throughput computing,” University of Illinois at Urbana-Champaign, Center for Reliable and High-Performance Computing, Technical Report IMPACT-12-01, 2012. [Online]. Available: <http://impact.crhc.illinois.edu/Shared/Docs/impact-12-01.parboil.pdf>.
- [14] A. Danalis, G. Marin, C. McCurdy, *et al.*, “The scalable heterogeneous computing (SHOC) benchmark suite,” in *Proceedings of the 3rd Workshop on General-Purpose Computation on Graphics Processing Units*, ser. GPGPU-3, Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2010, pp. 63–74, ISBN: 9781605589350. DOI: 10.1145/1735688.1735702.
- [15] S. Che, M. Boyer, J. Meng, *et al.*, “Rodinia: A benchmark suite for heterogeneous computing,” in *2009 IEEE International Symposium on Workload Characterization (IISWC)*, 2009, pp. 44–54. DOI: 10.1109/IISWC.2009.5306797.
- [16] K. Asanović, R. Bodik, B. C. Catanzaro, *et al.*, “The landscape of parallel computing research: A view from Berkeley,” Tech. Rep. UCB/EECS-2006-183, Dec. 2006. [Online]. Available: <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.html>.
- [17] S. Matsuoka, J. Domke, M. Wahib, *et al.*, “Preparing for the future—rethinking proxy applications,” *Computing in Science & Engineering*, vol. 24, no. 2, pp. 85–90, 2022. DOI: 10.1109/MCSE.2022.3153105.
- [18] G. Juckeland, W. Brantley, S. Chandrasekaran, *et al.*, “SPEC ACCEL: A standard application suite for measuring hardware accelerator performance,” in *High Performance Computing Systems. Performance Modeling, Benchmarking, and Simulation*, S. A. Jarvis, S. A. Wright, and S. D. Hammond, Eds., Cham: Springer International Publishing, 2015, pp. 46–67, ISBN: 978-3-319-17248-4. DOI: 10.1007/978-3-319-17248-4\_3.
- [19] S. Boehm, S. Pophale, V. G. Vergara Larrea, and O. Hernandez, “Evaluating performance portability of accelerator programming models using SPEC ACCEL 1.2 benchmarks,” in *High Performance Computing*, R. Yokota, M. Weiland, J. Shalf, and S. Alam, Eds., Cham: Springer International Publishing, 2018,

- pp. 711–723, ISBN: 978-3-030-02465-9. DOI: 10.1007/978-3-030-02465-9\_51.
- [20] N. Malaya, B. Messer, J. Glenski, *et al.*, “Experiences readying applications for exascale,” in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC ’23, Denver, CO, USA: Association for Computing Machinery, 2023, ISBN: 9798400701092. DOI: 10.1145/3581784.3607065.
- [21] PRACE, *UEABS: Unified European Applications Benchmark Suite*, <https://repository.prace-ri.eu/git/UEABS/ueabs>, Accessed: 2024-03-13, 2022.
- [22] *CORAL-2 benchmarks*, <https://asc.llnl.gov/coral-2-benchmarks>, Accessed: 2024-08-23.
- [23] *NERSC-10 benchmark suite*, <https://www.nersc.gov/systems/nersc-10/benchmarks/>, Accessed: 2024-03-22.
- [24] O. Pearce, A. Scott, G. Becker, *et al.*, “Towards collaborative continuous benchmarking for HPC,” in *Proceedings of the SC ’23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis*, ser. SC-W ’23, Denver, CO, USA: Association for Computing Machinery, 2023, pp. 627–635, ISBN: 9798400707858. DOI: 10.1145/3624062.3624135.
- [25] R. Adamson, P. Bryant, D. Montoya, *et al.*, “Creating continuous integration infrastructure for software development on DOE HPC systems,” *Computing in Science & Engineering*, pp. 1–9, 2024. DOI: 10.1109/MCSE.2024.3362586.
- [26] “TOP500 Nov 2023.” (Nov. 2023), [Online]. Available: <https://top500.org/lists/top500/2023/11/>.
- [27] D. Alvarez, “JUWELS Cluster and Booster: Exascale pathfinder with Modular Supercomputing Architecture at Juelich Supercomputing Centre,” *Journal of large-scale research facilities JLSRF*, vol. 7, A183, Oct. 2021, ISSN: 2364-091X. DOI: 10.17815/jlsrf-7-183.
- [28] S. Graf and O. Mextorf, “JUST: Large-scale multi-tier storage infrastructure at the Jülich Supercomputing Centre,” *Journal of large-scale research facilities JLSRF*, vol. 7, A180–A180, 2021. DOI: 10.17815/jlsrf-5-172.
- [29] K. Hoste, J. Timmerman, A. Georges, and S. De Weirdt, “EasyBuild: Building software with ease,” in *2012 SC Companion: High Performance Computing, Networking Storage and Analysis*, IEEE, 2012, pp. 572–582. DOI: 10.1109/SC.Companion.2012.81.
- [30] K. Haghighi Mood, A. Herten, and S. Achilles, *Jupiter benchmark suite: Amber*, 2024. DOI: 10.5281/ZENODO.12737559.
- [31] T. Hater, A. Herten, and J. Badwaik, *Jupiter benchmark suite: Arbor*, 2024. DOI: 10.5281/ZENODO.12737757.
- [32] E. B. Gregory, A. Herten, and S. Achilles, *Jupiter benchmark suite: Chroma-lqcd*, 2024. DOI: 10.5281/ZENODO.12737626.
- [33] J. H. Meinke, A. Strube, A. Herten, S. Achilles, and K. Haghighi Mood, *Jupiter benchmark suite: Gromacs*, 2024. DOI: 10.5281/ZENODO.12787776.
- [34] C. I. Meyer, N. Nobre Wittwer, M. Römmer, *et al.*, *Jupiter benchmark suite: Icon*, 2024. DOI: 10.5281/ZENODO.12787963.
- [35] D. Willsch, H. De Raedt, A. Herten, and S. Achilles, *Jupiter benchmark suite: Juqs*, 2024. DOI: 10.5281/ZENODO.12788076.
- [36] C. Witzler, J. Windgassen, M. Bode, A. Herten, and S. Achilles, *Jupiter benchmark suite: Nekrs*, 2024. DOI: 10.5281/ZENODO.12788254.
- [37] A. Gonzalez-Nicolas, D. Caviedes-Voullième, A. Strube, *et al.*, *Jupiter benchmark suite: Parflow*, 2024. DOI: 10.5281/ZENODO.12788364.
- [38] U. Sinha, J. Badwaik, A. Herten, and S. Achilles, *Jupiter benchmark suite: Picongpu*, 2024. DOI: 10.5281/ZENODO.12788381.
- [39] K. Haghighi Mood, A. Herten, and S. Achilles, *Jupiter benchmark suite: Quantum espresso*, 2024. DOI: 10.5281/ZENODO.12788398.
- [40] J. Badwaik, A. Smolenko, A. Herten, S. Achilles, and T. Breuer, *Jupiter benchmark suite: Soma*, 2024. DOI: 10.5281/ZENODO.12788506.
- [41] M. Cherti, S. Achilles, A. Herten, C. John, and J. Badwaik, *Jupiter benchmark suite: Mmoclip*, 2024. DOI: 10.5281/ZENODO.12788226.
- [42] C. John, S. Kesselheim, C. Penke, *et al.*, *Jupiter benchmark suite: Megatron-lm*, 2024. DOI: 10.5281/ZENODO.12788115.
- [43] C. John, A. Strube, S. Kesselheim, A. Herten, S. Achilles, and J. Ebert, *Jupiter benchmark suite: Resnet*, 2024. DOI: 10.5281/ZENODO.12788436.
- [44] K. Szabo, A. Herten, S. Achilles, J. Badwaik, and S. Nassyr, *Jupiter benchmark suite: Dynqcd*, 2024. DOI: 10.5281/ZENODO.12737680.
- [45] E. Behle, A. Herten, and S. Achilles, *Jupiter benchmark suite: Nastja*, 2024. DOI: 10.5281/ZENODO.12788527.
- [46] D. Alvarez, A. Strube, A. Herten, and S. Achilles, *Jupiter benchmark suite: Graph500*, 2024. DOI: 10.5281/ZENODO.12788553.
- [47] J.-O. Mirus, T. Breuer, S. Achilles, and A. Herten, *Jupiter benchmark suite: Hpcg*, 2024. DOI: 10.5281/ZENODO.12788585.
- [48] S. Achilles, A. Herten, and K. Haghighi Mood, *Jupiter benchmark suite: Hpl*, 2024. DOI: 10.5281/ZENODO.12788612.
- [49] Y. Müller, M. Holicki, A. Herten, and S. Achilles, *Jupiter benchmark suite: Linktest*, 2024. DOI: 10.5281/ZENODO.12788705.
- [50] T. Breuer, A. Herten, and S. Achilles, *Jupiter benchmark suite: Osu micro-benchmarks*, 2024. DOI: 10.5281/ZENODO.12788751.

- [51] S. Achilles, T. Breuer, K. Thust, *et al.*, *Jupiter benchmark suite: Stream*, 2024. DOI: 10.5281/ZENODO.12788781.
- [52] S. Achilles, A. Herten, and K. Haghighi Mood, *Jupiter benchmark suite: Stream*, 2024. DOI: 10.5281/ZENODO.12788801.
- [53] R. Salomon-Ferrer, A. W. Götz, D. Poole, S. Le Grand, and R. C. Walker, “Routine microsecond molecular dynamics simulations with AMBER on GPUs. 2. explicit solvent particle mesh Ewald,” *Journal of Chemical Theory and Computation*, vol. 9, no. 9, pp. 3878–3888, 2013. DOI: 10.1021/ct400314y.
- [54] D. A. Case, H. M. Aktulga, K. Belfon, *et al.*, “Amber-tools,” *Journal of Chemical Information and Modeling*, vol. 63, no. 20, pp. 6183–6191, 2023. DOI: 10.1021/acs.jcim.3c01153.
- [55] D. Cerutti. “Amber20 benchmarks.” (2020), [Online]. Available: [https://ambermd.org/Amber20\\_Benchmark\\_Suite.tar.gz](https://ambermd.org/Amber20_Benchmark_Suite.tar.gz).
- [56] S. F. Ashby and R. D. Falgout, “A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations,” *Nuclear science and engineering*, vol. 124, no. 1, pp. 145–159, 1996. DOI: 10.13182/NSE96-A24230.
- [57] J. E. Jones and C. S. Woodward, “Newton–Krylov-multigrid solvers for large-scale, highly heterogeneous, variably saturated flow problems,” *Advances in water resources*, vol. 24, no. 7, pp. 763–774, 2001. DOI: 10.1016/S0309-1708(00)00075-0.
- [58] J. Hokkanen, S. Kollet, J. Kraus, A. Herten, M. Hrynkiak, and D. Pleiter, “Leveraging HPC accelerator architectures with modern techniques — hydrologic modeling on GPUs with ParFlow,” *Computational Geosciences*, May 2021. DOI: 10.1007/s10596-021-10051-4.
- [59] L. Schneider and M. Müller, “Multi-architecture monte-carlo (mc) simulation of soft coarse-grained polymeric materials: Soft coarse grained monte-carlo acceleration (soma),” *Computer Physics Communications*, vol. 235, pp. 463–476, 2019, ISSN: 0010-4655. DOI: <https://doi.org/10.1016/j.cpc.2018.08.011>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0010465518303072>.
- [60] K. C. Daoulas and M. Müller, “Single chain in mean field simulations: Quasi-instantaneous field approximation and quantitative comparison with Monte Carlo simulations,” *The Journal of Chemical Physics*, vol. 125, no. 18, 2006. DOI: 10.1063/1.2364506.
- [61] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.
- [62] S. Borsanyi, Z. Fodor, J. N. Guenther, *et al.*, “Leading hadronic contribution to the muon magnetic moment from lattice qcd,” *Nature*, vol. 593, no. 7857, pp. 51–55, Apr. 2021, ISSN: 1476-4687. DOI: 10.1038/s41586-021-03418-1. [Online]. Available: <http://dx.doi.org/10.1038/s41586-021-03418-1>.
- [63] H. J. C. Berendsen, D. van der Spoel, and R. van Drunen, “GROMACS: A message-passing parallel molecular dynamics implementation,” *Computer Physics Communications*, vol. 91, no. 1, pp. 43–56, Sep. 1995, ISSN: 0010-4655. DOI: 10.1016/0010-4655(95)00042-E.
- [64] S. Páll, M. J. Abraham, C. Kutzner, B. Hess, and E. Lindahl, “Tackling exascale software challenges in molecular dynamics simulations with GROMACS,” in *Solving Software Challenges for Exascale*, ser. Lecture Notes in Computer Science 8759, S. Markidis and E. Laure, Eds., Springer International Publishing, Apr. 2014, pp. 3–27, ISBN: 978-3-319-15976-8. DOI: 10.1007/978-3-319-15976-8\_1.
- [65] M. J. Abraham, T. Murtola, R. Schulz, *et al.*, “GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers,” *SoftwareX*, vol. 1–2, pp. 19–25, Sep. 2015, ISSN: 2352-7110. DOI: 10.1016/j.softx.2015.06.001.
- [66] S. Páll, A. Zhmurov, P. Bauer, *et al.*, “Heterogeneous parallelization and acceleration of molecular dynamics simulations in GROMACS,” *The Journal of Chemical Physics*, vol. 153, no. 13, p. 134 110, Oct. 2020, ISSN: 0021-9606. DOI: 10.1063/5.0018516.
- [67] G. Zängl, D. Reinert, P. Rípodas, and M. Baldauf, “The ICON (ICOsahedral non-hydrostatic) modelling framework of DWD and MPI-M: Description of the non-hydrostatic dynamical core,” *Quarterly Journal of the Royal Meteorological Society*, vol. 141, no. 687, pp. 563–579, 2015. DOI: 10.1002/qj.2378.
- [68] M. Giorgetta, R. Brokopf, T. Crueger, *et al.*, “ICON-A, the atmosphere component of the ICON earth system model. part i: Model description,” *Journal of Advances in Modeling Earth Systems*, vol. 10, Jun. 2018. DOI: 10.1029/2017MS001242.
- [69] P. Korn, N. Brüggemann, J. H. Jungclaus, *et al.*, “ICON-O: The ocean component of the ICON earth system model—global simulation characteristics and local telescoping capability,” *Journal of Advances in Modeling Earth Systems*, vol. 14, no. 10, 2022. DOI: 10.1029/2021MS002952.
- [70] M. A. Giorgetta, W. Sawyer, X. Lapillonne, *et al.*, “The ICON-A model for direct qbo simulations on gpus (version icon-cscs:baf28a514),” *Geoscientific Model Development*, vol. 15, no. 18, pp. 6985–7016, 2022. DOI: 10.5194/gmd-15-6985-2022.
- [71] C. Hohenegger, P. Korn, L. Linardakis, *et al.*, “ICON-Sapphire: Simulating the components of the earth system and their interactions at kilometer and sub-kilometer scales,” *Geoscientific Model Development*, vol. 16, no. 2, pp. 779–811, 2023. DOI: 10.5194/gmd-2022-171.

- [72] B. Stevens, C. Acquistapace, A. Hansen, *et al.*, “The added value of large-eddy and storm-resolving models for simulating clouds and precipitation,” *Journal of the Meteorological Society of Japan. Ser. II*, vol. 98, no. 2, pp. 395–435, 2020. DOI: 10.2151/jmsj.2020-021.
- [73] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro, *Megatron-LM: Training multi-billion parameter language models using model parallelism*, 2020. arXiv: 1909.08053 [cs.CL].
- [74] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf).
- [75] D. Narayanan, M. Shoeybi, J. Casper, *et al.*, “Efficient large-scale language model training on GPU clusters using Megatron-LM,” in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC ’21, St. Louis, Missouri: Association for Computing Machinery, 2021, ISBN: 9781450384421. DOI: 10.1145/3458817.3476209.
- [76] V. Korthikanti, J. Casper, S. Lym, *et al.*, *Reducing activation recomputation in large transformer models*, 2022. arXiv: 2205.05198 [cs.LG].
- [77] T. Dao, *FlashAttention-2: Faster attention with better parallelism and work partitioning*, 2023. arXiv: 2307.08691 [cs.LG].
- [78] S. Rajbhandari, J. Rasley, O. Ruwase, and Y. He, “ZeRO: Memory optimizations toward training trillion parameter models,” in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC ’20, Atlanta, Georgia: IEEE Press, 2020, ISBN: 9781728199986.
- [79] C. M. John, J. Ebert, C. Penke, S. Kesselheim, and A. Hertel, “OpenGPT-X – Training Large Language Models on HPC Systems,” ISC High Performance 2023, Hamburg (Germany), 21 May 2023 - 25 May 2023, May 21, 2023. DOI: 10.34732/XDVBLG-SVNDMJ.
- [80] A. Radford, J. W. Kim, C. Hallacy, *et al.*, *Learning transferable visual models from natural language supervision*, 2021. arXiv: 2103.00020 [cs.CV].
- [81] C. Schuhmann, R. Vencu, R. Beaumont, *et al.*, “Laion-400m: Open dataset of clip-filtered 400 million image-text pairs,” *Data-Centric AI Workshop NeurIPS*, arXiv:2111.02114, 2021.
- [82] C. Schuhmann, R. Beaumont, R. Vencu, *et al.*, “LAION-5B: An open large-scale dataset for training next generation image-text models,” in *Thirty-sixth Conference on Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, vol. 35, 2022, pp. 25 278–25 294. [Online]. Available: <https://openreview.net/forum?id=M3Y74vmsMcY>.
- [83] G. Ilharco, M. Wortsman, R. Wightman, *et al.*, *Openclip*, version 0.1, Jul. 2021. DOI: 10.5281/zenodo.5143773.
- [84] M. Cherti, R. Beaumont, R. Wightman, *et al.*, “Reproducible scaling laws for contrastive language-image learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2818–2829.
- [85] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695.
- [86] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual instruction tuning,” in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [Online]. Available: <https://openreview.net/forum?id=w0H2xGHlkw>.
- [87] Q. Sun, Q. Yu, Y. Cui, *et al.*, “Emu: Generative pre-training in multimodality,” in *The Twelfth International Conference on Learning Representations*, 2023.
- [88] P. Giannozzi, S. Baroni, N. Bonini, *et al.*, “QUANTUM ESPRESSO: A modular and open-source software project for quantum simulations of materials,” *Journal of Physics: Condensed Matter*, vol. 21, no. 39, p. 395 502, Sep. 2009. DOI: 10.1088/0953-8984/21/39/395502.
- [89] P. Giannozzi, O. Andreussi, T. Brumme, *et al.*, “Advanced capabilities for materials modelling with Quantum ESPRESSO,” *Journal of Physics: Condensed Matter*, vol. 29, no. 46, p. 465 901, Oct. 2017. DOI: 10.1088/1361-648X/aa8f79.
- [90] P. Giannozzi, O. Baseggio, P. Bonfà, *et al.*, “Quantum ESPRESSO toward the exascale,” *The Journal of Chemical Physics*, vol. 152, no. 15, p. 154 105, Apr. 2020, ISSN: 0021-9606. DOI: 10.1063/5.0005082.
- [91] I. Carnimeo, F. Affinito, S. Baroni, *et al.*, “Quantum ESPRESSO: One further step toward the exascale,” *Journal of Chemical Theory and Computation*, vol. 19, no. 20, pp. 6992–7006, 2023. DOI: 10.1021/acs.jctc.3c00249.
- [92] *MaX benchmark repository*, [https://gitlab.com/max-centre/benchmarks/-/blob/master/Quantum\\_Espresso/CP/ZrO2/supercell\\_11layer/](https://gitlab.com/max-centre/benchmarks/-/blob/master/Quantum_Espresso/CP/ZrO2/supercell_11layer/).
- [93] *MaX: Materials at the eXascale. an EU Centre of Excellence for supercomputing applications*, <https://www.max-centre.eu/>, Accessed: 2024-03-14.
- [94] M. Berghoff and I. Kondov, “Non-collective scalable global network based on local communications,” in *2018 IEEE/ACM 9th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems (scalA)*, 2018, pp. 25–32. DOI: 10.1109/ScalA.2018.00007.
- [95] F. Graner and J. A. Glazier, “Simulation of biological cell sorting using a two-dimensional extended Potts

- model,” *Phys. Rev. Lett.*, vol. 69, pp. 2013–2016, 13 Sep. 1992. DOI: 10.1103/PhysRevLett.69.2013.
- [96] M. Berghoff, J. Rosenbauer, F. Hoffmann, and A. Schug, “Cells in silico – introducing a high-performance framework for large-scale tissue modeling,” *BMC bioinformatics*, vol. 21, no. 1, pp. 1–21, 2020. DOI: 10.1186/s12859-020-03728-7.
- [97] M. S. Steinberg, “On the mechanism of tissue reconstruction by dissociated cells, I. Population kinetics, differential adhesiveness, and the absence of directed migration,” *Proceedings of the National Academy of Sciences*, vol. 48, no. 9, pp. 1577–1582, 1962. DOI: 10.1073/pnas.48.9.1577.
- [98] N. A. Akar, B. Cumming, V. Karakasis, *et al.*, “Arbor — a morphologically-detailed neural network simulation library for contemporary high-performance computing architectures,” in *2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, 2019, pp. 274–282. DOI: 10.1109/EMPDP.2019.8671560.
- [99] K. Amunts, A. C. Knoll, T. Lippert, *et al.*, “The Human Brain Project—synergy between neuroscience, computing, informatics, and brain-inspired technologies,” *PLoS Biology*, vol. 17, no. 7, e3000344, 2019. DOI: 10.1371/journal.pbio.3000344.
- [100] Y. N. Billeh, B. Cai, S. L. Gratiy, *et al.*, “Systematic integration of structural and functional data into multi-scale models of mouse primary visual cortex,” *Neuron*, vol. 106, no. 3, 388–403.e18, 2020, ISSN: 0896-6273. DOI: <https://doi.org/10.1016/j.neuron.2020.01.040>.
- [101] R. G. Edwards and B. Joo, “The Chroma software system for Lattice QCD,” *Nucl. Phys. B Proc. Suppl.*, vol. 140, G. T. Bodwin, D. K. Sinclair, E. Eichten, *et al.*, Eds., p. 832, 2005. DOI: 10.1016/j.nuclphysbps.2004.11.254.
- [102] *USQCD*, <https://usqcd-software.github.io/>.
- [103] M. A. Clark, R. Babich, K. Barros, R. C. Brower, and C. Rebbi, “Solving Lattice QCD systems of equations using mixed precision solvers on GPUs,” *Comput. Phys. Commun.*, vol. 181, pp. 1517–1528, 2010. DOI: 10.1016/j.cpc.2010.05.002.
- [104] K. De Raedt, K. Michielsen, H. De Raedt, *et al.*, “Massively parallel quantum computer simulator,” *Comput. Phys. Commun.*, vol. 176, p. 121, 2007. DOI: 10.1016/j.cpc.2006.08.007.
- [105] H. De Raedt, F. Jin, D. Willsch, *et al.*, “Massively parallel quantum computer simulator, eleven years later,” *Comput. Phys. Commun.*, vol. 237, pp. 47–61, 2019. DOI: 10.1016/j.cpc.2018.11.005.
- [106] D. Willsch, M. Willsch, F. Jin, K. Michielsen, and H. De Raedt, “GPU-accelerated simulations of quantum annealing and the quantum approximate optimization algorithm,” *Comput. Phys. Commun.*, vol. 278, p. 108411, 2022. DOI: 10.1016/j.cpc.2022.108411.
- [107] F. Arute, K. Arya, R. Babbush, *et al.*, “Quantum supremacy using a programmable superconducting processor,” *Nature*, vol. 574, pp. 505–510, 2019. DOI: 10.1038/s41586-019-1666-5.
- [108] H. De Raedt and D. Willsch. “Jülich universal quantum computer simulator (docker container).” (2021), [Online]. Available: <https://jugit.fz-juelich.de/qip/juqcs-docker>.
- [109] H. De Raedt and D. Willsch. “Jülich universal quantum computer simulator (light version).” (2024), [Online]. Available: <https://jugit.fz-juelich.de/qip/juqcs-light>.
- [110] P. Fischer, S. Kerkemeier, M. Min, *et al.*, “NekRS, a GPU-accelerated spectral element Navier–Stokes solver,” *Parallel Computing*, vol. 114, p. 102982, 2022. DOI: 10.1016/j.parco.2022.102982.
- [111] E. Merzari, S. Hamilton, T. Evans, *et al.*, “Exascale multiphysics nuclear reactor simulations for advanced designs,” in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC ’23, Denver, USA: Association for Computing Machinery, 2023, ISBN: 9798400701092. DOI: 10.1145/3581784.3627038.
- [112] A. T. Patera, “A spectral element method for fluid dynamics: Laminar flow in a channel expansion,” *Journal of Computational Physics*, vol. 54, pp. 468–488, 1984. DOI: 10.1016/0021-9991(84)90128-1.
- [113] S. A. Orszag, “Spectral methods for problems in complex geometries,” *Journal of Computational Physics*, vol. 37, pp. 70–92, 1980. DOI: 10.1016/0021-9991(80)90005-4.
- [114] M. O. Deville, P. F. Fischer, and E. H. Mund, *High-Order Methods for Incompressible Fluid Flow*. Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 2002. DOI: 10.1017/CBO9780511546792.
- [115] D. S. Medina, A. St-Cyr, and T. Warburton, “OCCA: A unified approach to multi-threading languages,” *arXiv preprint arXiv:1403.0968*, 2014.
- [116] Hidden (Blind Review), “Boundary layers in thermal convection are fluctuation-dominated,” *Journal of Fluid Mechanics (submitted)*, 2024.
- [117] Hidden (Blind Review), “A high-scaling in-situ workflow for deciphering boundary layer effects in high-Rayleigh-number convection,” (*submitted*), 2024.
- [118] M. Bussmann, H. Burau, T. E. Cowan, *et al.*, “Radiative signatures of the relativistic Kelvin-Helmholtz instability,” in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, ser. SC ’13, Denver, Colorado: ACM, 2013, 5:1–5:12, ISBN: 978-1-4503-2378-9. DOI: 10.1145/2503210.2504564.
- [119] E. Zenker, B. Worpitz, R. Widera, *et al.*, “Alpaka – an abstraction library for parallel kernel acceleration,” in *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2016, pp. 631–640. DOI: 10.1109/IPDPSW.2016.50.
- [120] B. Worpitz, *Investigating performance portability of a highly scalable particle-in-cell simulation code on*

*various multi-core architectures*, Master Thesis, Sep. 2015. DOI: 10.5281/zenodo.49768.

- [121] H. Burau, R. Widera, W. Hönig, *et al.*, “PIConGPU: A fully relativistic particle-in-cell code for a GPU cluster,” *IEEE Transactions on Plasma Science*, vol. 38, no. 10, pp. 2831–2839, 2010. DOI: 10.1109/TPS.2010.2064310.
- [122] J. Kunkel, G. Markomanolis, J. Bent, and J. Lofstead, *Vi4io/io-500-dev: Zenodo citation release*, version v1.1, Sep. 2018. DOI: 10.5281/zenodo.1422814.
- [123] *LinkTest: Communication API benchmarking tool*, <https://gitlab.jsc.fz-juelich.de/cstao-public/linktest/>, Accessed: 2024-03-22.
- [124] *MaX3 CoE*, <https://www.max-centre.eu/>, Accessed: 2024-08-23.
- [125] *CoEC*, <https://coec-project.eu/>, Accessed: 2024-08-23.