UCLouvain

École polytechnique de Louvain

Parameter Identification of a Tumour Growth Model for Adaptive Radiotherapy

Author: **Nicolas Permanne**
Supervisor: **Benoît Macq**
Readers: **Mélanie Ghislain, Florian Martin, Manon Dausort, Ana Maria Barragan Montero**
Academic year 2023–2024
Master [120] in Computer Science and Engineering

Acknowledgements

Computational resources have been provided by the supercomputing facilities of the Université catholique de Louvain (CISM/UCL) and the Consortium des Équipements de Calcul Intensif en Fédération Wallonie Bruxelles (CÉCI) funded by the Fond de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under convention 2.5020.11 and by the Walloon Region

Firstly, I would like to thank my supervisor, Professor Benoit Macq. His insights and reflections have been a beneficial asset while developing my master thesis.

I wish to thank and give recognition to Manon Dausort, Mélanie Ghislain and Florian Martin. Their patience and kindness is truly inspiring and enabled me to conduct this research in a healthy and pleasant environment. From the initial discussions on project direction to the final review of the manuscript, their commitment and guidance greatly enhanced the quality of this work. Their constructive criticism enabled me to improve myself both personally and scientifically. The time they spent on this work has sincerely made a positive impact on both this master thesis and me.

I wish to extend my gratitude to Ana Maria Barragan Montero for her role as a reader and the time spend reviewing this master thesis.

I would like to acknowledge all individuals working at the Université catholique de Louvain. They helped to create and maintain a favourable environment for the personal development of students. I am profoundly appreciative of all the professors, PhD students, and tutors who have helped me achieve my goals during these years.

Finally, I would like to express my heartfelt thanks to my entourage. Family, friends, and partner have all been of an invaluable support during this work. Their love, advices, and confidence in my abilities have been instrumental through this period.

I would like to thank all of you for having accompanied, helped, and encouraged me during the creation of this master thesis. Without your presence, advices and support, this work would not have been possible. My sincere thanks to all you.

Abstract

Adaptive Radiation Therapy (ART) is a crucial medical technique in the field of Radiation Therapy (RT). It aims to dynamically adapt the treatment plan using systematic measurements. This master thesis investigates the possibility of creating an ART process in which a tumour growth model adjust itself to the real tumour anatomy. A new treatment plan can then be optimised using Reinforcement Learning (RL) as demonstrated by Moreau et al. [1] and Martin et al. [2].

Specifically, we address the inverse parameter identification problem on the combined cellular simulation proposed by Jalalimanesh et al. [3] and O'Neil et al. [4]. It is achieved by training a hybrid Neural Network (NN) architecture combining Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) on the output matrices of the simulation.

In practise, we are trying to predict five crucial initial parameters of the simulation: the duration of the cell cycle and the four average consumptions in both oxygen and glucose for both cancer and healthy cells. Our research shows that some of these parameters can be accurately predicted while others, which have less impact on the simulation, are more challenging to estimate. Notably, our most promising result is predicting the duration of the cell cycle with an error margin of within 5%.

Furthermore, due to the stochastic nature of the tumour growth model, we analyse the inherent similarity between simulations. This leads us to observe the consequences of the prediction error on the output matrices of the model.

This master thesis demonstrates the potential to dynamically adjust the parameters of a cellular model based on its outputs, leading to new prospects in the field of ART.

Contents

Acronyms

ABM Agent-Based Models. 16, 17, 20

AI Artificial Intelligence. 7, 9, 17

ART Adaptive Radiation Therapy. 2, 7, 9, 17, 49

CA Cellular Automata. 16, 17, 20

CNN Convolutional Neural Network. 2, 12, 13, 18, 26, 28, 35, 49, 50

CPM Cellular Potts Models. 16

CT Computed Tomography. 17

DNA Deoxyribonucleic Acid. 8, 9

EBRT External Beam Radiotherapy. 9

KNN K-Nearest Neighbour. 10, 49

LGCA Lattice-Gas Cellular Automata. 16, 20

LQ Linear Quadratic. 23

LSTM Long Short-Term Memory. 2, 11, 12, 13, 18, 28, 50, 71, 72

MAE Mean Absolute Error. 14

MDS Multi-Dimensional Scaling. 10

MI Mutual Information. 10, 35, 36, 38, 45, 48

ML Machine Learning. 9, 13

MRI Magnetic Resonance Imaging. 17

MSE Mean Square Error. 18, 30

NLP Natural Language Processing. 13

NN Neural Network. 2, 7, 11, 12, 13, 18, 26, 27, 31, 33, 37, 41, 43, 49, 50

NTCP Normal Tissue Complication Probability. 9, 17

OAR Organs At Risk. 17

ODEs Ordinary Differential Equations. 16, 17

OMF Oxygen Modified Factor. 23

PCA Principal Component Analysis. 9, 31, 32, 33, 34, 35

PCC Pearson Correlation Coefficient. 14

PDEs Partial Differential Equations. 16, 17

PET Positron Emission Tomography. 17

QA Quality Assurance. 17

ReLU Rectified Linear Unit. 11

RL Reinforcement Learning. 2, 7, 9, 26, 27, 37, 39, 44, 45, 46, 47, 48, 68

RMSE Root Mean Squared Error. 14

RNN Recurrent Neural Network. 12, 13, 18

RT Radiation Therapy. 2, 7, 9, 17, 26, 27, 38, 49, 50

SSIM Structural Similarity Index Measure. 15

SVM Support Vector Machine. 49

TCP Tumour Control Probability. 9, 17

Chapter 1

Introduction

Radiation Therapy (RT) is a medical technique used to control or destroy a tumour by irradiating cells. The specialist constructed a treatment plan that determines the specifications of the medical procedure.

The term Adaptive Radiation Therapy (ART) was introduced by Yan et al. in 1997 as a feedback-loop process that uses systematic measurements to continuously adapt the treatment plan. In the last decades, Artificial Intelligence (AI) has been a significant improvement in this field due to its ability to automate certain decisions and to process large amounts of data.

Parameter identification is the task of reconstructing the initial parameters of a system. It is often used on numerical models describing real-life processes with the purpose of retrieving the underlying parameters for a better calibration of the model.

In a recent study, Moreau et al. applied a Reinforcement Learning (RL) algorithm on a tumour growth simulation to optimise the treatment plan. This work was extended by Martin et al. in an effort to analyse the robustness of the RL algorithm under variations of the initial simulation parameters.

This master thesis aims at bringing the work of Moreau et al. and Martin et al. closer to an ART process. The long-term purpose is to dynamically adapt the tumour growth model to the real-life tumour anatomy. It is achieved by first translating medical imagery into the simulation and then retrieving the initial parameters to calibrate the simulation, and finally re-optimising the treatment plan. Our research focusses on the second part and aims at using a Neural Network (NN) to predict the initial parameters of the simulation based on its outputs.

In addition, due to the stochastic properties of the tumour growth model, we'll study the inherent similarity between simulations to assess the performance of our NN no only in terms of predicted parameters but also in terms of the difference they cause on the model.

This work is making its own contribution to a larger project that aims at using a tumour growth model based on the patient anatomy to deliver personalised RT treatments and ultimately advances the field of ART.

Chapter 2

Theoretical background

2.1 Radiotherapy

2.1.1 Cancer & tumour

A tumour is the result of an anomalous growth of cells in the body. They are trillions of cells in the human body and a tumour can develop itself from any of these cells. Typically, cells go through the cell cycle which consists of 4 phases [5]:

1. **The G_1 phase**: The G stands for gap and over this duration, the cell grows and produces essentials materials such as proteins, ribosomes and important organelles for cells that rely on them. This phase takes the most time and lasts about 11 hours.
2. **The S phase**: This period is called the synthesis and it is used to duplicate its Deoxyribonucleic Acid (DNA). This phase lasts about 8 hours.
3. **The G_2 phase**: Like in the G_1 phase, the cell produces a lot of proteins. The cell also checks if the DNA is correctly replicated and that no defect has been made. This phase lasts about 4 hours.
4. **The M phase**: This period is called the mitosis and it is during this time that the actual division from a parent cell into 2 daughters cells occurs. This phase lasts about 1 hour.

After mitosis, some cells do not divide anymore and enter a quiescence state, the G_0 phase. This phenomenon can happen if a cell is not meant to produce daughter cells (neurons for example), if the density of cells around is too high or if there's not enough nutrients to ensure the survival of the cells.

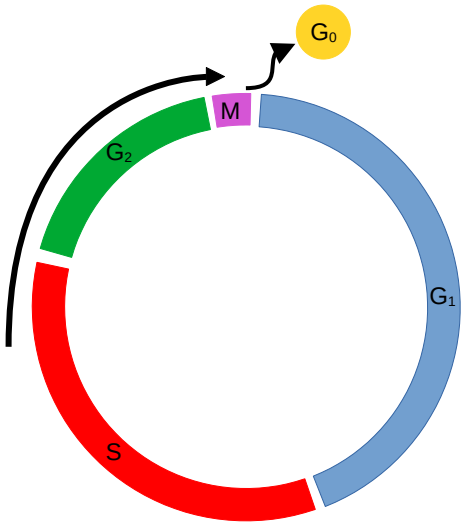

Figure 2.1: Different steps constituting the cell cycle

Errors can occur during these different steps and especially during the S phase leading to genetic defects in the cell and potentially to a tumour. If the tumour becomes invasive or spreads to neighbouring tissues via the blood and lymphatic system, it is known as a malignant tumour or cancer. Benign tumours also exist and can even be quite large. Cancer cells have differences compared to healthy cells [6]:

- Unlike healthy cells, cancer cells do not need a signal to grow and ignore signals telling them to stop dividing or die. This phenomenon is called apoptosis.
- Cancer cells invade nearby regions and spread across the body whereas healthy cells stop growing when approaching other types of cells. It is generally described under the name of metastasis.
- Cancer cells instruct blood vessels to expand near the tumour to supply themselves with nutrients and oxygen. We call this angiogenesis.
- When healthy cells are damaged, the immune system removed them through the lymphocytes. Cancer cells are able to evade these lymphocytes by avoiding their detection or by secreting chemicals that inhibit them.

2.1.2 Basic principle of radiation therapy

Radiation Therapy (RT) is a medical technique used to control or destroy a tumour by irradiating cells. The radiation used are of different kind but in the perspective of this work, the employed method is External Beam Radiotherapy (EBRT) based on beams of photons, protons or electrons [7]. These particles have a wide variety of effects on cells, but the purpose is usually to cause direct damages to DNA in order to kill the cell or deteriorate its reproducing capability. Unfortunately, these effects can also impact neighbouring healthy cells. Therefore, when planning a RT treatment, we aim at maximising the effect of the radiation on tumoural cells while minimising their impact of healthy cells. This optimization problem can be formulated through two metrics: the Tumour Control Probability (TCP) used to calculate the percentage of tumour killing and the Normal Tissue Complication Probability (NTCP) that models the impact of radiation on neighbouring healthy cells. The patient receives a dose of radiation expressed in Grays (Gy) with one gray corresponding to one joule of radiation energy per kilogram of matter. This dose is divided into fraction and spread in time, this is known as dose fractionation.

The machine delivering the radiation contains a particle accelerator that goes along with an imaging technology and a treatment planning program helping the technician in his work. This last part can use the data acquired by the machine to adapt the plan in real time, this is called Adaptive Radiation Therapy (ART)(section 3.2)

2.2 Machine learning

Machine Learning (ML) is a field of study in Artificial Intelligence (AI) that covers the set of techniques and algorithms able to express some kind of learning. This domain is divided into 3 main categories:

- Supervised learning has for purpose to learn a specific function for which we already have many points and their target value. If the target is continuous it's a regression, on the contrary it's a classification for categorical data.
- Unsupervised learning does not have target value and tries to find meaning on unlabelled data. This includes task like clustering that regroup the point in the dataset into multiples groups (cluster). Another task is the dimensionality reduction that transforms the data from a high-dimensional space to a low-dimensional space while keeping some useful insight.
- Reinforcement Learning (RL) aims at taking intelligent decisions in a specific environment. The RL agent takes actions on the environment that in return gave feedback to the agent so that it can improve himself.

2.2.1 Dimensionality reduction

Dimensionality reduction is an unsupervised learning method that transform data from an original high-dimensional space to a low-dimensional output space while keeping meaningful properties of the data. It is often used for two purposes:

- To observe datasets beforehand to discover some of their underlying structures. In this case, the output space of the method uses only a few dimensions (often two or three for easier visualisation).
- As a pre-processing step when working with high dimensional data. It helps prevent the curse of dimensionality which is a group of problems arising when the number of dimensions is high and generally higher than the number of data points.

Principal Component Analysis

Principal Component Analysis (PCA) is the easiest to understand and popular dimensional reduction method. It works by projecting the data on the axis carrying maximal information about these data. The chosen axis is the one maximizing the variance of the projected data among all possibles axis passing through the centroid of the cloud of points. We can iteratively apply PCA and impose an orthogonality constraint between principal components to obtain a new coordinates system.

The optimization problem to be solved aims at maximizing the variance while keeping the norm of the direction vector of the transformation $\mathbf{v}$ equal to 1 and is formulated along with its Lagrange function as:

$$\max_{\mathbf{v}}(\mathbf{v}^T\Sigma\mathbf{v}) \text{ subject to } (\mathbf{v}^T\mathbf{v} = 1) \quad \Rightarrow \quad \mathscr{L} = \mathbf{v}^T\Sigma\mathbf{v} + \lambda(1 - \mathbf{v}^T\mathbf{v}) \tag{2.1}$$

With Σ the variance-covariance matrix, $\mathbf{v}$ the direction vector of the transformation and $\mathscr{L}$ the Lagrange function.

When $\frac{\partial \mathscr{L}}{\partial \mathbf{v}} = 0$ it gives the following eighen system:

$$\Sigma \mathbf{v} = \lambda \mathbf{v} \quad \Leftrightarrow \quad \lambda = \mathbf{v}^T \Sigma \mathbf{v} \quad \Leftrightarrow \quad \lambda = \sigma_{\mathbf{v}}^2 \tag{2.2}$$

The eighen vector $\mathbf{v}$ corresponding the largest eighen value λ is to the unit direction vector of the transformation into the principal component.

Isometric mapping

Isomap or Isometric mapping is a non-linear dimensionality reduction technique that aims at keeping the inherent geometry of the high dimensional data manifold. A manifold is a type of space that is locally similar to a euclidean space. The geodesic distance represents the distance of the shortest path between two points on this manifold. Isomap is based on the idea that the high dimensional space to reduce is a manifold and its purpose is to preserve the geodesic distance between point.

In practice, the algorithm operates in the following way:

1. The euclidean distance is computed between each point and the K-Nearest Neighbour (KNN) are calculated.
2. Based on these KNN we create a graph where an edge is created between two neighbouring points, the weight of the edge is the euclidean distance between the point.
3. The geodesic distance between each point of the graph is given through a shortest path algorithm like Dijkstra or Floyd-Warshall.
4. Finally, Multi-Dimensional Scaling (MDS) is used to transform these distances into individuals points.

2.2.2 Mutual Information

Mutual Information (MI) is a measure of the relationship between two variables. It is often understood as a kind of correlation able to express the non-linearity between the two variables. In more formal terms, the MI between two random variables is the amount of information obtained about one variable by observing the other.

Mathematically, for two random variables X and Y, the mutual information $I(X,Y)$ quantifies how the uncertainty on Y is reduced when X is known and vice versa. The uncertainty of a random variable can be measured with its entropy $H(X)$ which itself can be interpreted as the average number of bits (or other information units) needed to describe observations of the random variable:

$$\begin{aligned} H(X) &= -\mathrm{E}\left[\log(p(X))\right] \\ &= -\sum_{x \in \mathcal{X}} p(x) \log(p(x)) \end{aligned} \tag{2.3}$$

Conditional entropy $H(Y|X)$ measures the uncertainty on Y when X is known: $H(Y|X) = H(Y,X) - H(X)$.

Finally, the mutual information is expressed as the difference between the entropy of Y and the entropy of Y when X is known:

$$\begin{aligned} I(X,Y) &= H(X) - H(X|Y) = H(Y) - H(Y|X) \\ &= \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} P_{(X,Y)}(x,y) \log\left(\frac{P_{(X,Y)}(x,y)}{P_X(x) P_Y(y)}\right) \end{aligned} \tag{2.4}$$

Mutual information is often used when selecting the features before applying a model, it quantifies two requirements:

- The relevance of the features is measured by the mutual information between the features and the target(s).
- The redundancy of the features is measured by the mutual information between themselves.

We want to point out that despite being often compared with the correlation, the MI is not normalised and could take any positive value. When doing a relevance analysis, we can normalised the mutual information by dividing it by the entropy of the target which gives us a result between 0 and 1. For ease of comprehension, we'll refer to this measure as the normalised mutual information.

2.2.3 Neural network

A Neural Network (NN) is the machine learning method at the core of deep learning. It approximates a function mainly for supervised learning tasks even if it can be applied to other types of machine learning. NN are inspired by biological neural network and are consequently composed of many small interconnected units called neurons. These neurons work like a non-linear regression and are able to model complex function when put together.

The function to be approximated with a domain of size N is:

$$y : \mathbb{R}^N \to \mathbb{R} \text{ with } P \text{ observed values } \{y_i, x_{i,1}, x_{i,2}, ..., x_{i,N-1}, x_{i,N}\}_{i=0}^{P} \tag{2.5}$$

$\widehat{y}$ is the approximation of y following (see Figure 2.2):

$$\widehat{y}(\mathbf{x}) = \phi(\mathbf{w} \cdot \mathbf{x}) \text{ with } \mathbf{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_N \end{pmatrix} \text{ and } \mathbf{x} = \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_N \end{pmatrix} \tag{2.6}$$

Where ϕ is the activation function adding non-linearity to the approximation and $\mathbf{w}$ the weight vector. Here are some of the most popular activation functions:

- Linear function ($f(x) = x$): this corresponds to no activation function, barely used.
- Sigmoid/logistic function ($f(x) = \frac{1}{1+e^{-x}}$): constrained the output between zero and one, useful in Long Short-Term Memory for example (subsection 2.2.5).
- Hyperbolic tangent ($f(x) = \tanh(x)$): constrained the output between minus one and one, useful in Long Short-Term Memory for example (subsection 2.2.5).
- Rectified Linear Unit (ReLU) ($f(x) = \max(0, x)$): with its variant, it's one of the most used activation function.

The error of approximation is described using a loss function (also known as criterion). This function is to be minimised by the learning process of the NN.

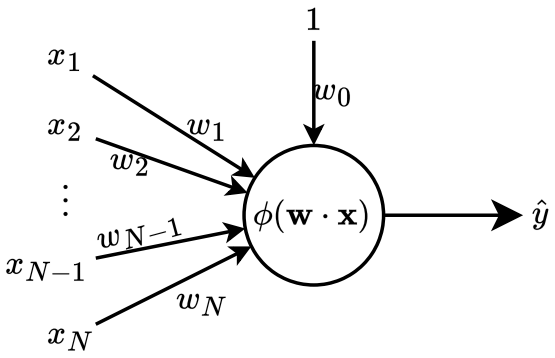

Figure 2.2: Schematic of a typical neuron in a NN

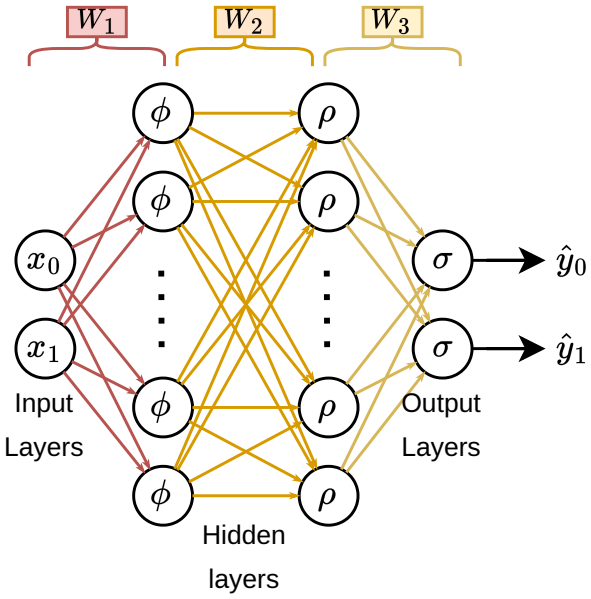

Figure 2.3: Example of a NN with one input layer, 2 hidden layers and one output layer

A Neural Network is composed of an agglomeration of neurons into layers, the layers between the input and output layer are the hidden layers. For example, the function that describes the neural network in Figure 2.3 is:

$$\widehat{y}_k = \sigma\left(\mathbf{W}_3 \cdot \rho\left(\mathbf{W}_2 \cdot \phi\left(\mathbf{W}_1 \cdot \mathbf{x}\right)\right)\right) \tag{2.7}$$

The weight of these models (the vectors $\mathbf{w}_i$ and the matrices $\mathbf{W}_i$) are learned through an iterative process called " Gradient Descent " that computes the gradient of the loss function and adjust the weight following this gradient:

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \alpha \left.\frac{\partial Loss}{\partial \mathbf{w}}\right|_{\mathbf{w}(t)} \tag{2.8}$$

with α the learning rate. The gradient of the loss function with respect to the weights can be computed using the backpropagation algorithm based on the chain rule formula. For three functions f, g and h, the chain rule is:

$$h = (f \circ g) \Rightarrow h'(x) = (f' \circ g) \cdot g' \quad \text{or, equivalently,} \quad \frac{\partial h}{\partial x} = \frac{\partial f}{\partial g} \cdot \frac{\partial g}{\partial x} \tag{2.9}$$

The **backpropagation algorithm** works by iterating backward over all the layers, it efficiently avoids redundant calculations by reusing results from the layer previously calculated. For this motive, the loss function and all the activation functions must be differentiable.

The gradient descent algorithm is run on individual block of data called batch, the size of these batches is a hyperparameter controlling the trade-off between accuracy and speed. A larger batch size allows learning multiple data at once but may result in lower accuracy. One epoch or iteration is done when all data have been passed through the NN.

The **vanishing and exploding gradient problem** are two problems occurring as the NN grows in layers. During the training process when we used the backpropagation algorithm to compute the gradient, the chain rule causes this gradient to be multiplied repeatedly as we iterate backward leading to a significant increase in the magnitude of the gradient. This is known as the exploding gradient problem. On the contrary, when using activation function with small derivatives such as the sigmoid, the magnitude tends to zero as we iterate backward. It is called the vanishing gradient problem. These problems are often associated with RNN leading to the creation of LSTM subsection 2.2.5.

2.2.4 Convolutional neural network

A Convolutional Neural Network (CNN) is a special type of Neural Network having the ability to learn features from grid-like data such as images. The neuron applies a convolution between the image and a filter (also known as kernel) that is used as weight being learned during the training process.
The mathematical operation behind the convolution is a cross-correlation corresponding to a sliding dot product between the filter and sub-matrices of the image Figure 2.4. The cross-correlation is defined here:

$$(f \star g)[m,n] = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f[m,n] g[m+i, n+j] \tag{2.10}$$

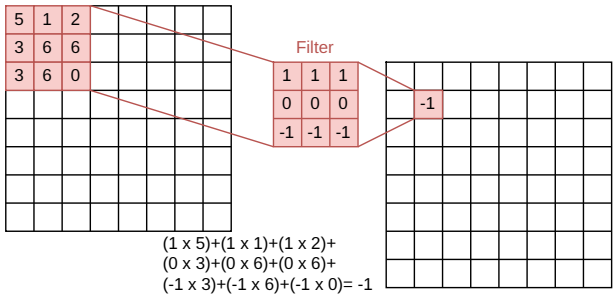

Figure 2.4: Example of convolution between an image and a filter

Filters are composed of 4 hyperparameters that can be adjusted to have different behaviours:

- Kernel size (k): Size of the filter/kernel.
- Padding (p): Numbers of pixels padded with zeros outside the input matrices.
- Dilation (d): It's how far the different pixels of the kernel are separated.
- Stride (s): Step size the kernel moves across the input data.

The output dimensions of the image are influenced by these parameters following this formulation:

$$Output = \frac{Input + 2 \cdot p - d \cdot (k-1) - 1}{s} + 1 \tag{2.11}$$

They are several applications of CNN such as image compression, image segmentation and image classification. In many architecture (and also in this work), CNN are used to extract relevant features in the image for further processing by

the remaining of the network. A typical CNN pipeline is composed of alternating convolutions and pooling layers with a normal feed-forward layers at the end (see Figure 2.5). Max(or average) pooling layers take the maximum (or the average) values in a sliding window over the image; this operation allows the reduction in size of the image further improving the efficiency and reducing the risk of overfitting.

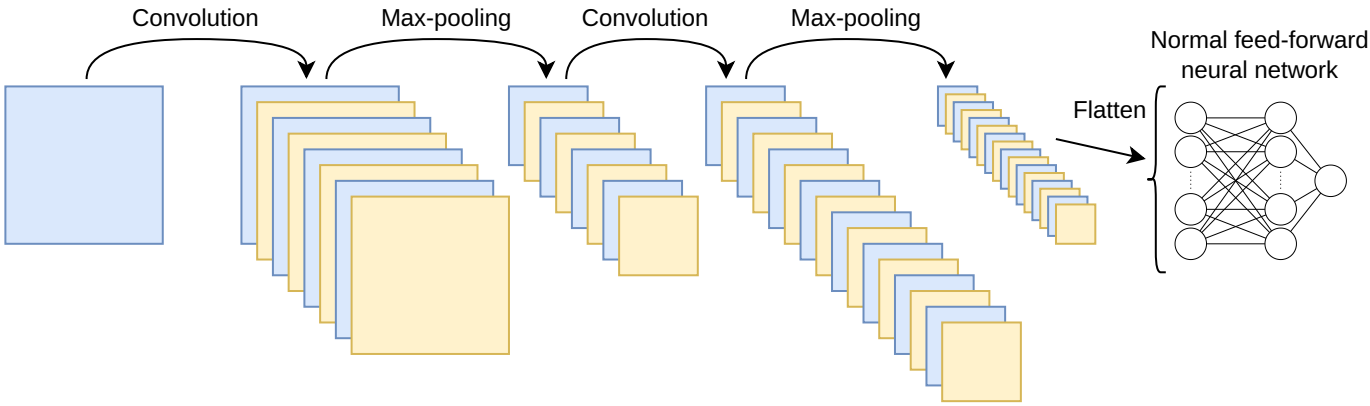

Figure 2.5: Example of a typical CNN architecture

2.2.5 Long short-term memory

Long Short-Term Memory (LSTM) networks are a kind of Recurrent Neural Network (RNN) which are themselves a kind of neural network used for sequential data and times-series. RNN allows for some outputs of the neurons to be the input of subsequent neurons in the same layer and consequently to have a flow of information between the nodes of the same layer. This addition to a traditional NN significantly impacts the vanishing/exploding gradient problem subsection 2.2.3 leading to the creation of LSTM. LSTM solves partially the vanishing gradient problem by introducing gating capabilities to control the flow of information and consequently the gradient. Another improvements over RNN is the addition of a long-term memory lasting for multiples timesteps.

LSTM are very useful for classification, prediction, and others classic ML task of data based on time series. It was widely used in Natural Language Processing (NLP) before the utilization of transformers.

A LSTM layer is divided into multiples blocks. Figure 2.6 depicts one block which is composed of three parts:

1. First, the hidden state, responsible for the short term memory, combined with the input predicts what percentage of the long term memory (called the cell state) need to be preserved. This part is known as the forget gate.
2. Then, the input gate is used to compute the quantity to be added to the cell state.
3. The output gate computes the new hidden state based on the newly calculated cell state.

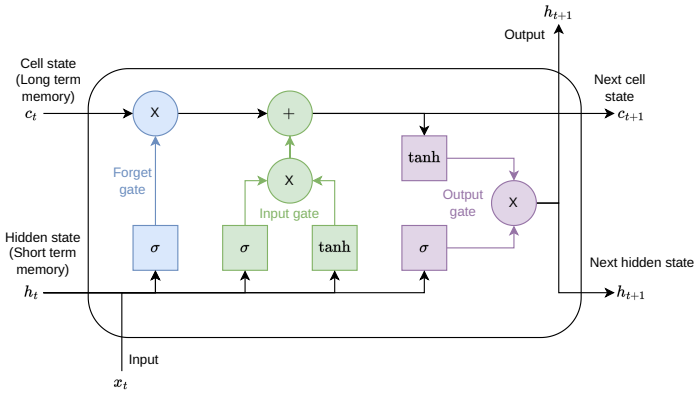

Figure 2.6: Schematic of the inner principle of a LSTM block

2.3 Metrics for matrices similarity

The simulation used and described in this master's thesis is based on multiples matrices. In order to compare efficiently these simulations between each other, we need to use metrics able to express the similarity between two matrices. Both dissimilarity and similarity metrics are described in this section, as one is often easily transform into the other. Mathematically, we represent a dissimilarity measure or difference by a function $S(A, B)$ that obeys the following properties [8]:

1. $S(A, B) \geq 0$

2. $S(A, B) = 0$ if $A = B$
3. $S(A, B) = S(B, A)$

Similar properties also exists for similarity metrics.

The following metrics are inspired by the literature on the subject [8], [9], [10].

2.3.1 L-p distance

This first family of dissimilarity metrics is derived from the l^p-norm of the difference:

$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}. \tag{2.12}$$

For the L-1 norm, we obtain the Manhattan distance, which is equivalent to the Mean Absolute Error (MAE):

$$mae(\mathbf{A}, \mathbf{B}) = \|\mathbf{A} - \mathbf{B}\|_1 := \frac{1}{mn} \sum_{i,j} |a_{i,j} - b_{i,j}|. \tag{2.13}$$

With m and n the dimensions of the matrices.

For the L-2 norm, we obtain the Euclidean distance, which is equivalent to the Root Mean Squared Error (RMSE):

$$rmse(\mathbf{A}, \mathbf{B}) = \|\mathbf{A} - \mathbf{B}\|_2 := \sqrt{\frac{1}{mn} \sum_{i,j} (a_{i,j} - b_{i,j})^2} \tag{2.14}$$

For the L-∞ norm, we obtain the Chebyshev distance which is equivalent to the maximum absolute error:

$$D_{Chebyshev}(\mathbf{A}, \mathbf{B}) = \|\mathbf{A} - \mathbf{B}\|_\infty := \max_{i,j} |a_{i,j} - b_{i,j}| \tag{2.15}$$

2.3.2 Cosine Similarity and Pearson's correlation

The cosine similarity measures the cosines of the angle between two non-zeros vectors and acts as a similarity metric. Mathematically, it is formulated as:

$$S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i,j} a_{i,j} b_{i,j}}{\sqrt{\sum_{i,j} a_{i,j}^2} \cdot \sqrt{\sum_{i,j} b_{i,j}^2}} \tag{2.16}$$

The Pearson Correlation Coefficient (PCC) is a statistical measure of the linear correlation between a pair of random variables. It is defined as the ratio between the covariance between the variables and the product of their standard deviation:

$$\rho_{X,Y} = \frac{\operatorname{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{\mathbb{E}[X\,Y] - \mathbb{E}[X]\,\mathbb{E}[Y]}{\sqrt{\mathbb{E}[X^2] - (\mathbb{E}[X])^2}\,\sqrt{\mathbb{E}[Y^2] - (\mathbb{E}[Y])^2}} \tag{2.17}$$

For two samples of data and specifically for our image, we obtain:

$$S(A, B) = \frac{\sum_{i,j}(a_{i,j} - \bar{a})(b_{i,j} - \bar{b})}{\sqrt{\sum_{i,j}(a_{i,j} - \bar{a})^2} \sqrt{\sum_{i,j}(b_{i,j} - \bar{b})^2}} \tag{2.18}$$

If our data is already centred, cosine similarity and PCC produces the same measure. These metrics range from -1 to 1. 1 means perfect correlation, 0 no correlation at all and -1 a perfect anti-correlation.

2.3.3 Mutual Information

Mutual information has already been discussed in subsection 2.2.2 and can be used to describe how much information is shared between both images to establish a value of similarity between them.

2.3.4 Structural Similarity Index Measure

The Structural Similarity Index Measure (SSIM) is a metric that encompasses three important keys features to compare images: the luminance, the contrast and the structure [11].

The luminance is computed with the average of all pixel values. The comparison function is expressed as:

$$l(\mathbf{A}, \mathbf{B}) = \frac{2\mu_A\mu_B + C_1}{\mu_A^2 + \mu_B^2 + C_1} \tag{2.19}$$

The constant C_1 is used to ensure the stability when the denominator approaches zeros.

The contrast is measured based on the standard deviation of the pixels values. Its comparison function is:

$$c(\mathbf{A}, \mathbf{B}) = \frac{2\sigma_A\sigma_B + C_2}{\sigma_A^2 + \sigma_B^2 + C_2} \tag{2.20}$$

The structure is computed with the following formulation: $\mathbf{x} - \mu_x/\sigma_x$ and it is compared using:

$$s(\mathbf{A}, \mathbf{B}) = \frac{\sigma_{AB} + C_3}{\sigma_A\sigma_B + C_3} \tag{2.21}$$

Finally, these three functions are multiplied together and weighted using the parameter α, β and γ:

$$SSIM(\mathbf{A}, \mathbf{B}) = [l(\mathbf{A}, \mathbf{B})]^\alpha \cdot [c(\mathbf{A}, \mathbf{B})]^\beta \cdot [s(\mathbf{A}, \mathbf{B})]^\gamma \tag{2.22}$$

We note that this computation is usually done locally on windows of the image and then unify in one metric by taking the average.

2.3.5 Histogram-based similarity

These metrics first computes the histogram of the image which is the empirical distribution of the pixel intensities:

$$hist_I(l) = \frac{\#\{(i,j) : I(i,j) = l\}}{m \cdot n} \tag{2.23}$$

with $I(i,j)$ the intensity of the image at pixel i, j.

From the two histograms obtained, we can compute their correlation and use the result as our similarity measure:

$$\text{histogram correlation}(A, B) = \rho_{hist_A, hist_B} \tag{2.24}$$

We also compute the intersection between the two histograms and normalised it:

$$\text{histogram intersection}(A, B) = \frac{\sum_{i=1}^{n} \min\{hist_A, hist_B\}}{\sum_{i=1}^{n} hist_A} \tag{2.25}$$

2.3.6 Set-based similarity metrics

The 2 followings metrics are defined on discrete sets of data. Both can be expressed in terms of operation on binary vectors. In our work, we extend these metrics to function on vector with more than two class. We achieve this by using boolean masks and computing the mean of the metrics for all the masks.

The Dice-Sørensen coefficient is defined as :

$$Dice(A, B) = \frac{2\,|A \cap B|}{|A| + |B|} \underset{generalized}{\Longrightarrow} \frac{2\,|A \cdot B|}{|A|^2 + |B|^2} \tag{2.26}$$

The Jaccard similarity coefficient, also known as the Tanimoto coefficient is defined as the ratio of the size of the intersection over union:

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \underset{generalized}{\Longrightarrow} \frac{|A \cdot B|}{|A|^2 + |B|^2 - |A \cdot B|} \tag{2.27}$$

Chapter 3

Literature Review

3.1 Cancer simulation

Cancer simulations are algorithms that aim to replicate biological, chemical, and physical processes, determining how healthy and tumoural cells act in the human body. These simulations are called "in silico" models in opposition to "in vivo" describing experiments done inside an organism and "in vitro" describing experiments in laboratory or test tube.

Mathematical models of cancer are useful to understand tumour behaviour and the mechanics behind their growth. These simulations lead to new experiments, different treatment modalities, and altered prognostic risks. The impact of these simulations can be explained by their ability to reveal unknown or overlooked principles through qualitative approach of biology [12].

Models of cancer are numerous and have been explored by different fields of study: statistical, probabilistic, game theory and physics-based models. Edelman et al. [13] sort their examples by computational approach with statistical models, network-based models, and finally models on tissue level. We base our work on tissue level models categorised based on three general techniques: continuum models, discrete models, and hybrid combining both continuum and discrete ones.

Continuum models use a system of Partial Differential Equations (PDEs) and Ordinary Differential Equations (ODEs) to describe changes in cell population and environment. ODEs are responsible for explaining only temporal variations happening in the cell, while PDEs explain both spatial and temporal variations. Solution to these systems of equations can be evaluated through numerical techniques such as finite-element methods and are less computationally intensive compared to discrete and hybrid models [14]. They are particularly suitable to study the macroscopic scale of the biological system and can be directly compared to experimental data. The most known continuum model is the reaction-diffusion model that has been used in part for radiotherapy.

In **discrete individual-cell-based models**, each cell is a unique entity with its own properties and variables. A big difference with continuum models is the intrinsic stochastic nature of the simulation. Discrete simulations describe cell growth, migration and other biological phenomena based on the probabilities of these events. For example, vessels movement and angiogenesis are usually described using a biased random walk. Discrete models require biophysical parameters that impact the growth of the tumour, these parameters include diffusion of growth factors, hormones, nutrients, and oxygen.

Cells can either be fixed on a grid, we say that the model is on-lattice, or freely moving, and the model is off-lattice. On-lattice models can be categorised further by spatial resolution [15]:

- Cellular Automata (CA) contain a single cell per pixel (or voxel in three dimensions) on the grid, they are ideally used at a macroscopic scale.
- Lattice-Gas Cellular Automata (LGCA) contain multiples cells per pixels, they are ideally used at mesoscopic and macroscopic scales.
- Cellular Potts Models (CPM) use multiples pixels to represent one cell, they are ideally used at a mesoscopic scale.

The term Agent-Based Models (ABM) is common in the tumour simulation literature and is generally used as another way to say discrete individual-cell-based models. However, Anvari et al. [14], it is used to describe a specific off-lattice model. In this work, we'll use the definition from Jeanquartier et al. [16]: "ABM depicts a computational method for simulating a system, which is based on individual units, calculated by a given rule-set on a discrete level".

Hybrid discrete-continuum models regroup the advantages of both continuum and discrete models into a highly complex description of the tumour dynamics. This combination allows them to make microscale descriptions while also being efficient for macroscopic scale characteristics. In general, they combine systems of PDEs and ODEs with ABM or CA.

Altrock et al. [12] also looked at cancer initiation and the dynamics of mutation. Mathematical models for these tasks have been used to predict risk of cancer based on factors like age or smoking history. Powerful models in this category includes branching and Moran processes that are both Markov processes using the assumption that cellular events do not influence each other.

3.2 Adaptive radiation therapy

Radiation Therapy is a crucial technique in the cancer treatment. Traditionally it was used on the assumption that the anatomical properties of the tumour and of the body did not vary in time. For this reason, treatments were planned based on initial scans of the patient and the few adaptations during treatment involved the addition of margins to targets [17].

In the last decades, improvements in RT technologies have allowed a more precise delivery of radiation. Medical imagery such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI) and Positron Emission Tomography (PET) scans have made it easier to observe the changes in anatomy. The alteration of the body affects both the tumour and the healthy organs. They also occur on different time scales: from seconds for cardiac and respiratory motion to weeks for treatment-related variation like weight loss or tumour regression [18].

While techniques have improved to become more accurate and precise, a shift in paradigm was needed for further advancements in this field leading to Adaptive Radiation Therapy (ART). The concept of ART was introduced in 1997 by Yan et al. for lung cancer [19]. It is defined as a feedback-loop process that uses systematic measurements to continuously adapt the treatment plan. The purpose is to deposit doses to kill cancer cells or slow down their proliferation (TCP) while minimising the damage dealt to neighbouring cells (NTCP) and especially to Organs At Risk (OAR).

One of the important improvements leading to ART is AI. It allows to lighten the workload of radiation oncology specialists by automating certain decisions about customized beam arrangement, beam number, geometry, intensity, and modulation [17]. An AI algorithm is able to handle the rapidly increasing amount of data obtained from patient monitoring. Non-exclusive tasks that AI can solve include image segmentation, deformable registration, automated treatment planning, error detection, automated decision-making, and Quality Assurance (QA).

Three general approach to ART exist:

- The online approach adjusts the treatment just before fractions. This approach requires frequent visualization and powerful AI algorithms to adapt the treatment. For this reason, it is a more complex and ressource-intensive method. The dose administration is usually applied on a daily or weekly basis with the adjustments done at each session. It is employed when anatomical changes are expected on a daily basis such as abdominal and pelvic malignancies.
- The offline approach adjusts the treatment between fractions. Due to the way this approach operates, treatments are usually applied on a larger timescale with a lower frequency of radiation administration session. This additional time allows using less complex algorithm to analyse and adapt the treatment. It is more suited for patient with tumour and OAR less prone to high variation in time and when the tumour is far away from critical organism structure [17].
- Inline or real-time adjust the treatment during a fraction to account for variations such as respiration, internal status changes, or peristalsis motion. Real-time ART is performed by the treatment machine to align the radiation beam on the target [20].

3.3 Inverse parameter identification for image time-series

Parameter identification is the task of the reconstruction of the initial parameters of a system. This system is often a numerical simulation based on real-life processes and the purpose is to find the underlying parameters for a better calibration of the system. In general, these numerical simulations are based on a system of PDEs and ODEs and the task is an inverse problem because we somehow try to invert the operation of solving these equations [21]:

- The direct problem consists of evaluating the function F at x: $F(x) = y$
- The inverse problem consists of determining x from given data y: $y = F(x)$

In this work, the simulations follow a dual stochastic-deterministic nature (section 4.1 and section 5.2). Indeed, while the cell cycle is described by a rule-based scheme, many intrinsic factors are determined randomly, inducing dissimilarities between simulations involving same initial parameters. For this reason, the reciprocal of the function F, F^{-1} does not exist but may still be estimated with some inherent errors.

Meißner et al. [22] describes the 2 mains methods used for parameter identification:

- Iterative optimisation is the most commonly used algorithm for this task. It defines an error function between parameterised model outputs and actual outputs. This error is iteratively minimised using an optimisation algorithm. This method suffers from a high computational cost due to the need for new simulation every usage.
- Direct inverse implies the existence of an inverse relation between outputs and inputs. This relation is then approximated by a model such as a neural network.

Most of the numerical simulations vary in time. Therefore, time-series are a natural approach to consider in this task. A time-series at his core is just a succession of data points order in time. One of the main field of studies in this topic is the time-series analysis. It consists of the analysis of the time-series to infer interesting characteristics such as time-varying parameters, for example this is done in [23] for a tumour growth model using moving horizon estimation. A similar task is achieved in our work with a succession of images, called an image time-series. The three following paragraph extend on techniques to analyse these time-series.

A method often used for time-series analysis is the LSTM (see subsection 2.2.5). Sun et al. [24] created a NN architecture to produce land cover maps, they employed LSTM to take advantage of the temporal features present in satellite images times-series. A similar research was carried out [25] where a LSTM model outperformed another RNN but also both a CNN architecture and a support vector machine algorithm. LSTM are also involved in the case of a parameter identification task to retrieve parameters of transmission lines [26].

CNN are another type of NN useful in the parameter identification task due to their ability to extract meaningful features from data (see subsection 2.2.4). They are often used on time-series by either using a temporal, one dimensional CNN [27] or by transforming the time-series into an image before using the CNN to analyse and extract information relevant to the task [28].

Parameter identification tasks are commonly used on time-series but rarely on image time-series due to the added complexity of two dimensions over one. One field of study that stands out in terms of image time-series is the video study area. Due to the same type of input data, techniques and NN architectures can potentially be adapted to a parameter identification task. Wu et al. [29] used a hybrid model for semantic video classification. Their model is composed of CNN layers to extract short-term motion and spatial features, then LSTM are employed to model long-term clues before prediction. Similar architecture is employed in [30] to identify the parameters of an elastic-plastic constitutive model. CNN, LSTM and the combination of both is compared for the detection and classification of disturbances on power grid [31], the hybrid model CNN - LSTM outperformed the two others. The NN architecture used in this work is inspired by these CNN - LSTM hybrid models.

Inverse parameter task shares common characteristics in the literature. The presence of a similar processing pipeline is often described [22], [26], [30]. Figure 3.1 depicts a simple summary of these pipelines. The predicted parameters by the NN are normalised between their respective range to avoid bias toward specific parameters [26], [30]. It is usually done in the interval $[0, 1]$ or $[-1, 1]$ following the formulation:

$$y = \frac{(y_{max} - y_{min})(\xi - \xi_{min})}{(\xi_{max} - \xi_{min})} + y_{min} \tag{3.1}$$

with y the normalised parameter and ξ the original parameter. We retrieve the non-normalised version with the inverse formulation:

$$\xi = \frac{(\xi_{max} - \xi_{min})(y - y_{min})}{(y_{max} - y_{min})} + \xi_{min} \tag{3.2}$$

Finally, the loss function employed is the MSE [22], [26], [30], [32] due to its ability to increase proportionally to the square of the prediction error.

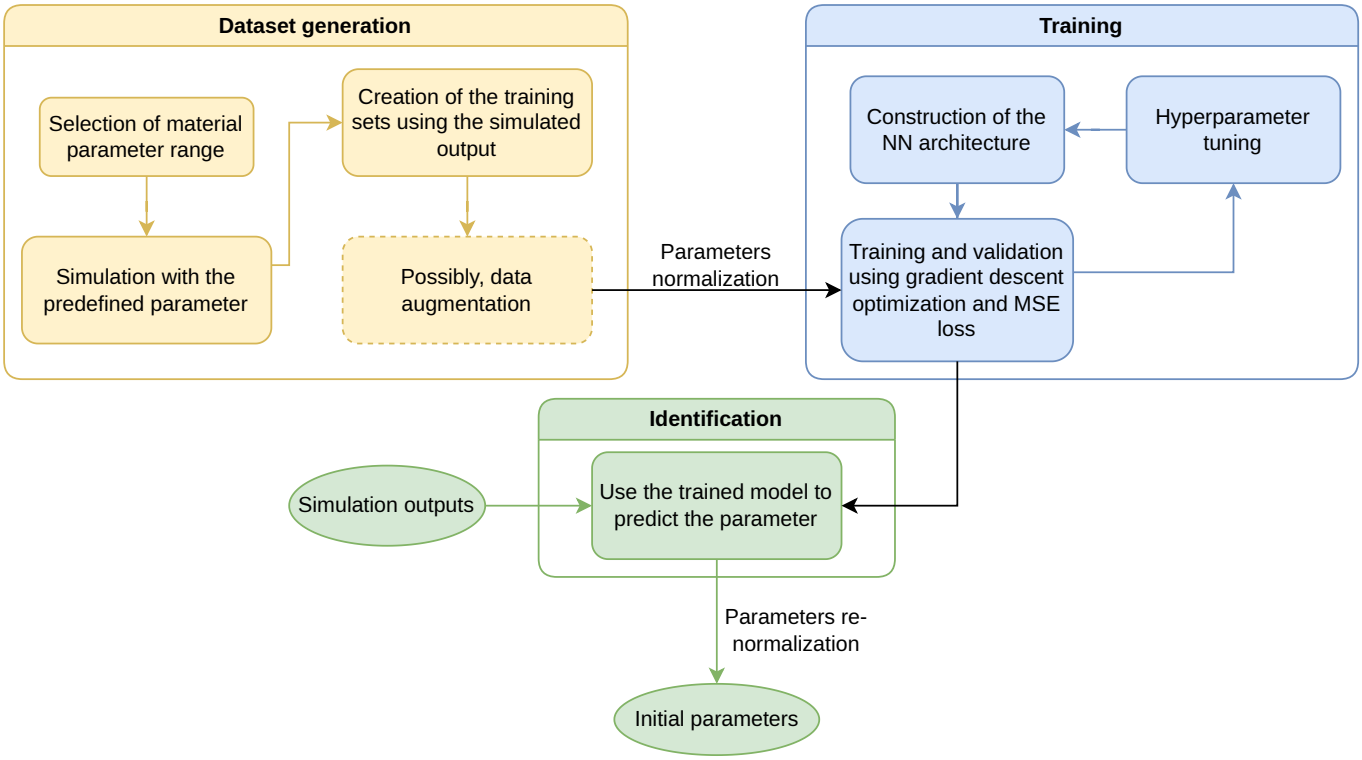

Figure 3.1: Schematic of the general processing pipeline of an inverse parameter identification task

Chapter 4

Materials & Methods

4.1 Simulation

This master's thesis follows the work of Moreau et al. [1] and Martin et al. [2] and consequently studies the same simulation. This simulation is taken from Jalalimanesh et al. [3] who extend the model from O'Neil et al. [4]. This section describes and explains the simulation.

The mathematical model used in this simulation is a discrete cell-based model. It evolved from a CA into an ABM. We would define it as a LGCA (see section 3.1). The lattice is defined as a 50-by-50 grid (adapted to 64-by-64 in this work) with each pixel containing a list of cells called a patch, nutrients (oxygen and glucose) and potentially a source of nutrients to simulate angiogenesis. Variables and properties of the cells are updated each hour following the chart in Figure 4.1. If all goes well, cells go through each phase of the cell cycle (see subsection 2.1.1) spending a certain time t_{phase} to finally divide after phase M. During the G_1 phase, if the surrounding cell density is too high or nutrients are insufficient, the cell transitions to a quiescent state known as the G_0 phase. On the contrary, during the G_0 phase, if the cell density is not too high, and if the nutrients are sufficient, the cell transitions to phase G_1. This quiescent state is implemented only for healthy cells to represent the defective growth control of tumoural cells. Cells in this state consumes $\frac{3}{4}$ the quantity of nutrients in other states. Each hour that passes, the cell consumes oxygen and absorbs glucose from the environment. In each phase, if glucose and oxygen levels are insufficient and the cell reaches critical thresholds ($q_{g,critical}$ and $q_{o,critical}$), the cell dies and is removed from the simulation.

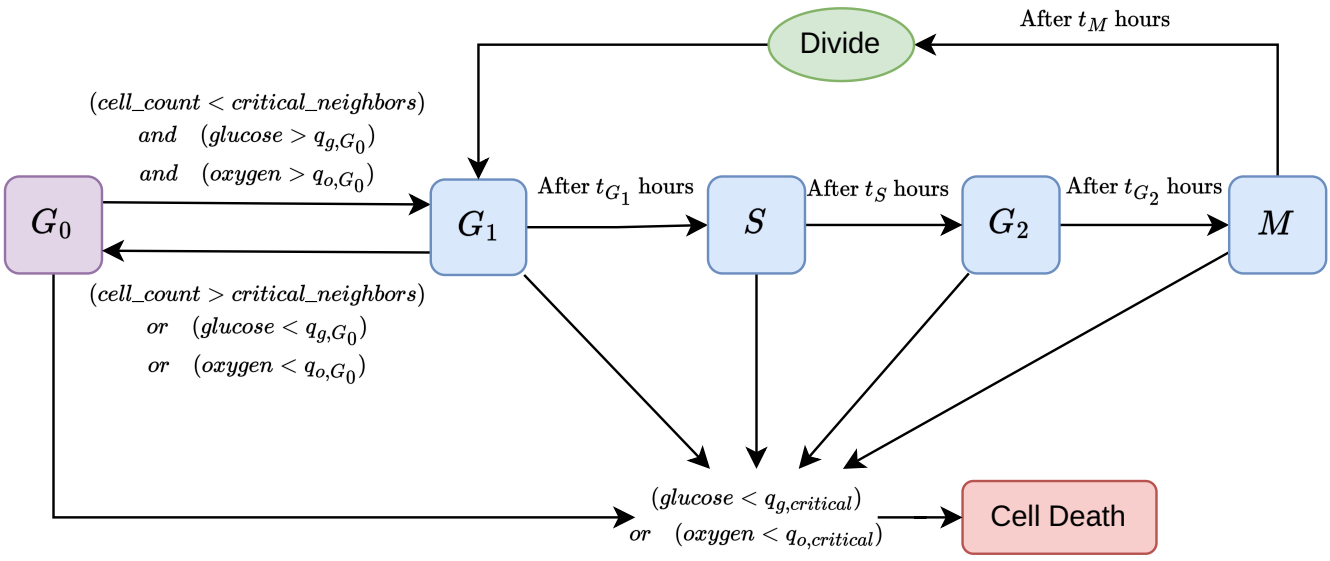

Figure 4.1: Cell cycle flow chart

4.1.1 Duration of the cell cycle

The duration of the cell cycle can vary a lot in function of the cell type, but for a typical eukaryotic cell, this duration is around 24 hours [33] divided into four phases. For this simulation, the durations of the phases are presented in Table 4.1.

Parameter	Description	Value
t_{G_1}	Duration of the G_1 phase	12
t_S	Duration of the synthesis phase	6
t_{G_2}	Duration of the G_2 phase	4
t_M	Duration of the mitosis phase	2
t_{total}	Total duration of the cell cycle	24

Table 4.1: Duration of the four phases of the cell cycle

4.1.2 Nutrients consumption

O'Neil et al. [4], defined glucose as the main nutrient of the simulation. They associated an average absorption of $\mu_{g,healthy} = 3.6 \times 10^{-9} mg/cell/hour$ for healthy cells and $\mu_{g,tumour} = 5.4 \times 10^{-9} mg/cell/hour$. Jalalimanesh et al. [3] added the oxygen to the simulation. They found that the average consumption of it is equal to $\mu_{o,healthy} = \mu_{o,tumour} = 2.16 \times 10^{-9} ml/cell/hour$ for both healthy and tumoural cells.

To add variations in the simulation, the actual quantity of nutrients is randomized in the form of an efficiency value. For both glucose and oxygen, this value is defined as the average value multiplied by a factor randomly determined on a normal distribution with a mean of 1 and a standard deviation of $\frac{1}{3}$. This factor is kept between 0 and 2 to avoid negative consumption of nutrients and to limit nutrient absorption to twice the average value:

$$efficiency = \mu \cdot \max\left(0, \min\left(2, \mathcal{N}\left(1, \frac{1}{3}\right)\right)\right). \tag{4.1}$$

This efficiency is computed at the creation of the cell for healthy cells and each hour for tumoural cells.

Healthy cells in a quiescence state consume fewer nutrients, their consumption at each hour is defined as $\frac{3}{4}$ the efficiency of healthy cells in another phase.

Parameter	Description	Value	Scaled Value
$q_{g,ini}$	Starting glucose	$1 \times 10^{-6}\ mg$	100
$q_{o,ini}$	Starting oxygen	$1 \times 10^{-6}\ ml$	1000
$\mu_{g,healthy}$	Average absorption of glucose for healthy cells	$3.6 \times 10^{-9}\ mg/cell/hour$	0.36
$\mu_{g,tumour}$	Average absorption of glucose for tumoural cells	$5.4 \times 10^{-9}\ mg/cell/hour$	0.54
$\mu_{o,healthy}$	Average consumption of oxygen for healthy cells	$2.16 \times 10^{-9}\ ml/cell/hour$	20
$\mu_{o,tumour}$	Average consumption of oxygen for tumoural cells	$2.16 \times 10^{-9}\ ml/cell/hour$	20

Table 4.2: Specific value of the parameters for simulation of nutrients

4.1.3 Critical and quiescence levels

The quiescence level is the threshold quantity for which cells enter the quiescence state (G_0) from the G_1 phase. It has been defined as the average quantity of nutrients required for two healthy cells to complete an entire cell cycle [4]:

$$\begin{aligned} q_{g,G_0} &= 2 \cdot t_{total}\,(hours) \cdot 3.6 \cdot 10^{-9}\,(mg/cell/hour) = 1.728 \cdot 10^{-7}\,(mg/cell) \\ q_{o,G_0} &= 2 \cdot t_{total}\,(hours) \cdot 2.16 \cdot 10^{-9}\,(ml/cell/hour) = 10.37 \cdot 10^{-8}\,(ml/cell)\,. \end{aligned} \tag{4.2}$$

The critical level is the minimum quantity of nutrients needed for cells to stay alive. It has been defined as the average quantity of nutrients required for one healthy cell in quiescence during one cell cycle:

$$\begin{aligned} q_{g,critical} &= \tfrac{3}{4} \cdot t_{total}\,(hours) \cdot 3.6 \cdot 10^{-9}\,(mg/cell/hour) = 6.48 \cdot 10^{-8}\,(mg/cell) \\ q_{o,critical} &= \tfrac{3}{4} \cdot t_{total}\,(hours) \cdot 2.16 \cdot 10^{-9}\,(ml/cell/hour) = 3.88 \cdot 10^{-8}\,(ml/cell)\,. \end{aligned} \tag{4.3}$$

It is important to note that the critical level should always be above the average nutrient consumption of cancer cells. In the case if this condition is not respected, the amount of nutrient can be lower than the critical level, therefore allowing

the cell to stay alive, while having the average nutrient consumption higher than the actual amount of nutrient leading to a negative value for the amount of nutrient which is not accommodated by the simulation. This essential condition is mathematically formulated as:

$$\begin{aligned} q_{critical} &> \mu_{cancer} \\ \lambda_{critical} \cdot t_{total} \cdot \mu_{healthy} &> \mu_{cancer}. \end{aligned} \tag{4.4}$$

This will prove useful when determining parameters ranges (see subsection 4.2.1).

4.1.4 Angiogenesis and nutrients diffusions

To simulate angiogenesis, $n_{sources}$ of sources of nutrients are placed at random on the grid. These sources move using a random walk approach with a higher probability of moving toward the tumour proportional to the number of cells in this tumour. Each of these sources deposits a quantity of oxygen and glucose (q_o and q_g) on the grid that will spread uniformly at a rate d_c following the differential equation:

$$c_j^{t+1} = (1 - d_c)c_j^t + \frac{1}{8} \sum_{k \in N(j)} d_c c_k^t \tag{4.5}$$

with c_i^t the nutrients' concentration on patch i at time t and $N(j)$ the neighbouring path of j.

Parameter	Description	Value	Scaled value
$d_{c,oxygen}$	Rate of diffusion of oxygen	0.2	0.2
$d_{c,glucose}$	Rate of diffusion of glucose	0.2	0.2
$n_{sources}$	Number of sources of nutrients	100	100
q_0	Supply of oxygen at the source	4.86×10^{-7} $ml/source/hour$	4500
q_g	Supply of glucose at the source	1.3×10^{-6} $mg/source/hour$	130

Table 4.3: Specific values of the parameters for simulation of angiogenesis and nutrients diffusions.

4.1.5 Radiation therapy in the simulation

Radiation

The effect of radiation is included in the simulation in the form of an algorithm that irradiates the tumour for a specific dose. The algorithm starts by computing the centre of mass of the tumour with the average of the position of the different cells. Then, the radius of the tumour is determined as the longest distance to a tumoural cell. This information is then used to compute the coverage radius of the dose. It depends on the distance from the centre and is defined as a convolution between a Gaussian and a rectangular window. Figure 4.2 depicts the whole process with the centre, the radius and the percentage of dose for each pixel.

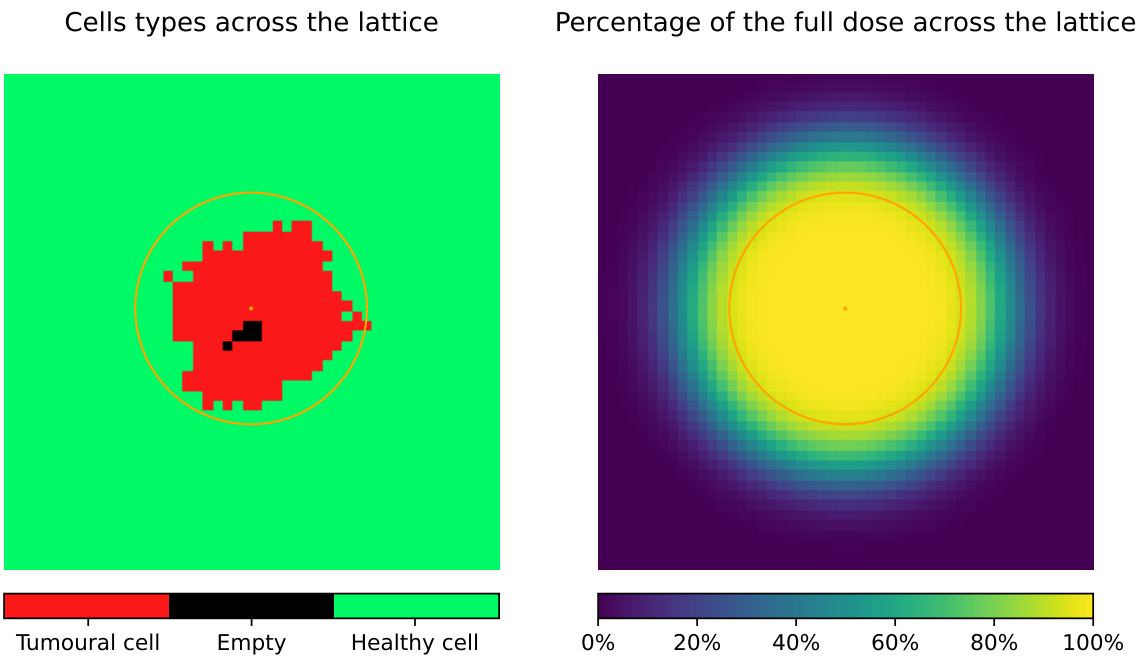

Figure 4.2: Example of dose delivery for a tumour with the centre and radius in orange

Response to radiation

A Linear Quadratic (LQ) function is used to model the radiation response. It describes the surviving fraction (SF) of cells exposed to a dose d in Grays (Gy):

$$SF(d) = \exp(-\alpha d - \beta d^2) \tag{4.6}$$

α and β are expressed respectively in Gy^{-1} and in Gy^{-2}. They are dependent of the radiosensitivity of the tissue and are specific to the tumour type.

Jalalimanesh et al. [3] used a modified LQ function to describe the survival probability of a cell irradiated with a dose d:

$$S(d) = \exp[\gamma_r(-\alpha \cdot OMF \cdot d - \beta(OMF \cdot d)^2)]. \tag{4.7}$$

The Oxygen Modified Factor (OMF) is a variable that symbolises the fact that hypoxic cells (cells deprived of the adequate oxygen quantity) are less sensitive to radiation. Finally, γ_r is a parameter that ranges from 0 to 1 controlling the sensitivity to radiation for the different phases of the cell cycle. Values of γ_r have been subject to some explorations leading to values above 1 as seen in Table 4.4.

Between each irradiation, cells have the ability to repair themselves. Each cell has a repair counter that resets to zero once the cell is fully repaired. Every hour, cells undergo repair by decreasing their repair counter by one. They can only advance in age and progress through the cell cycle once they are completely repaired. When irradiated, if the dose received is above a specific threshold (0.5), the repair counter is incremented by a random value between zero and two times the repair time of the cell.

Parameter	Value
$\alpha_{healthy}$	0.15 Gy^{-1}
$\beta_{healthy}$	0.03 Gy^{-2}
α_{tumour}	0.3 Gy^{-1}
β_{tumour}	0.03 Gy^{-2}

Parameter	Value
γ_{G_1}	1
γ_S	0.75
γ_{G_2}	1.25
γ_M	1.25
γ_{G_0}	0.75

Table 4.4: Specific value of the parameters for simulation of radiotherapy treatment

4.1.6 Simulation behaviours

This simulation, even if it does not capture all the biological and physical mechanisms that a real tumour may have, is quite complex and leads to interesting behaviours. This section aims at studying some of these behaviours to help us understand and interpret our results later.

Evolution of the simulation with respect to the duration of the cell cycle t_{total}

The duration of the cell cycle is an essential parameter in the simulation. Figure 4.3 depicts the direct influence of it on the tumour growth, the shorter the duration the faster the tumour grows and spreads. The consequence of this growth is an overpopulation of cancer cells: the centre of the tumour does not have enough nutrients and these cells therefore start to necrotise while the edge of the tumour is still developing. We observe that this phenomenon already occurs at the start for short cell cycle whereas longer simulations perceive it until much later and with less severity.

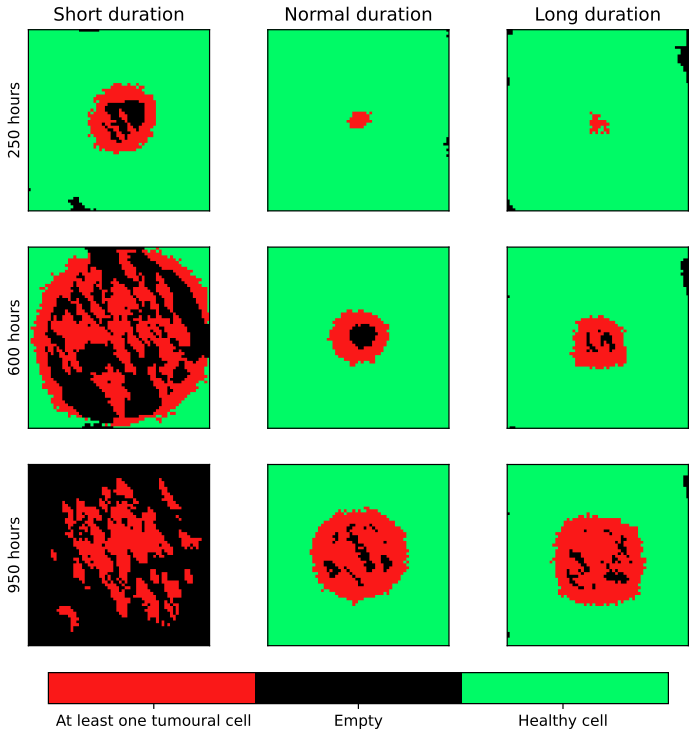

Figure 4.3: Temporal evolution of the tumour for short($12h$), normal($24h$) and long($36h$) duration of the cell cycle.

We add that even if Figure 4.3 does not depict a significant difference between normal and longer duration, the observation of the cells' density shows a lower amount of cells when the duration is longer.

Evolution of the simulation with respect to the average nutrients consumptions of cancers cells ($\mu_{g,cancer}$ and $\mu_{o,cancer}$)

When studying these average nutrients' consumption, we observed that simulation changes occur essentially on nutrients'density matrices. Figure 4.4 depicts these observations for the average oxygen consumption of cancer cells $\mu_{o,cancer}$. We noticed a lack of nutrients translating to a depletion zone in the centre of the simulation, at the precise location of the tumour. The consequence is described in the previous section (see section 4.1.6).

The first phenomenon to be observed is the increase in intensity of this oxygen-depleted zone, related to a higher average oxygen consumption of the tumour $\mu_{o,cancer}$. The second phenomenon is the opposite of the first for the glucose density. We observed that when the oxygen consumption is larger, the intensity of the glucose depletion zone diminishes. These phenomenons also appear for the glucose consumption $\mu_{g,cancer}$, and they can be formulated as:

$$\begin{array}{lll} \uparrow \mu_{o,cancer} \implies & density_o \downarrow \text{ and } & density_g \uparrow \\ \uparrow \mu_{g,cancer} \implies & \underbrace{density_g \downarrow}_{\text{First phenomenon}} \text{ and } & \underbrace{density_o \uparrow}_{\text{Second phenomenon}} \end{array} \tag{4.8}$$

While the first phenomenon is easily understood, the second one involves a more complex interpretation. We interpret that the oxygen and glucose consumptions counterbalance each other in a way. Our hypothesis is that when the tumour consumes more oxygen and consequently a lack of it is observed, cancer cells start dying faster due to hypoxia leading eventually to a diminution of the glucose density. On the contrary, if there is less oxygen consumed by the tumour, it will produce more cells leading to a total higher absorption of glucose and a more important depletion zone of the glucose. This hypothesis is supported by observations of cells' density.

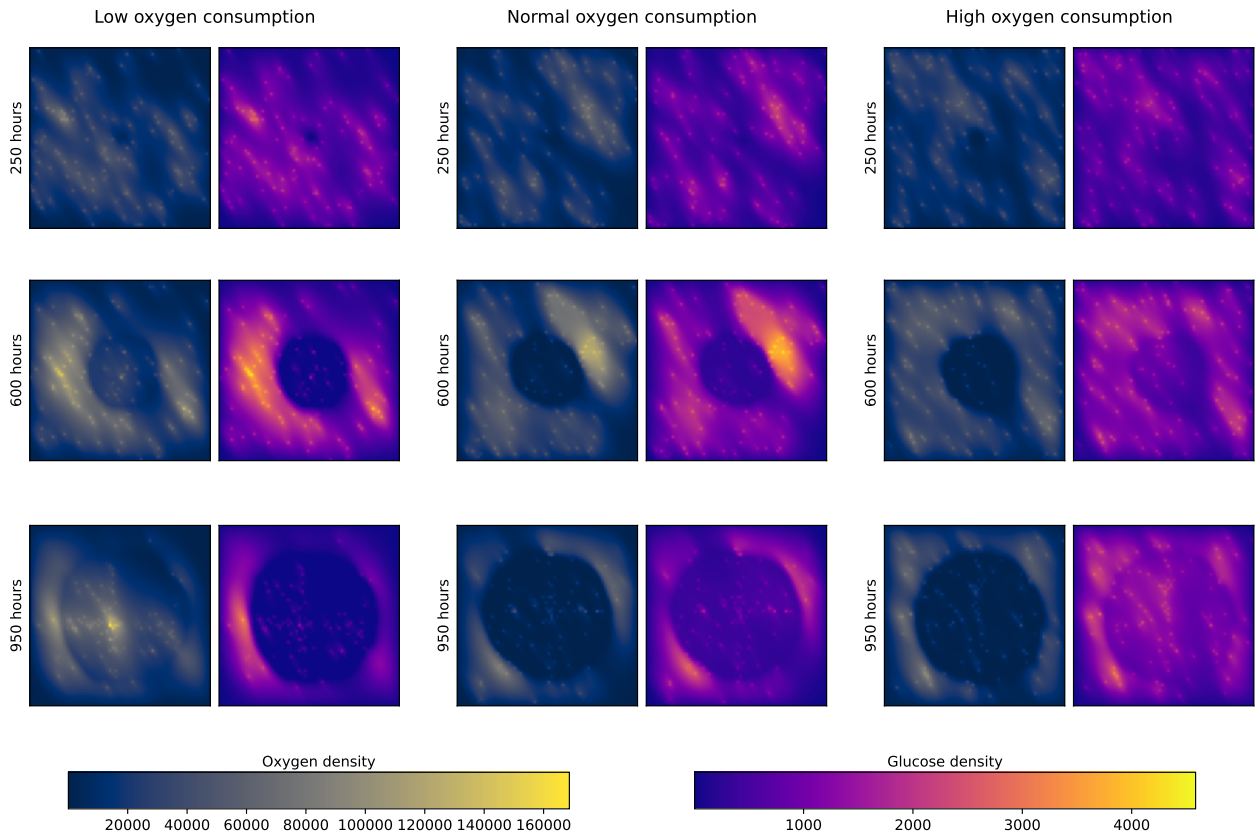

Figure 4.4: Temporal evolution of the density of glucose and oxygen for low, normal and high average oxygen consumptions by cancer cells

4.2 Dataset generation

The inverse model identifying the starting parameters based on the simulation output needs to be trained on a wide variety of different simulations. This section describes in more detail the datasets used and how they are generated.

The dataset extracts four kinds of matrices from the simulation:

- Cell type matrix: each element takes only three possibles values:

$$e_{ij} = \begin{cases} 0 & \text{if no cells is present on pixel } i,j \\ -1 & \text{if at least one cancer cell is present on pixel } i,j \\ 1 & \text{in other case} \end{cases}$$

- Cell density matrix: contains the number of cells on each pixel of the grid.
- Glucose density matrix: contains the amount of glucose on each pixel of the grid.
- Oxygen density matrix: contains the amount of oxygen on each pixel of the grid.

The choice of these matrices is quite arbitrary and other types of matrices could also be considered, they correspond to the observation of the RL algorithm used by Moreau et al. [1].

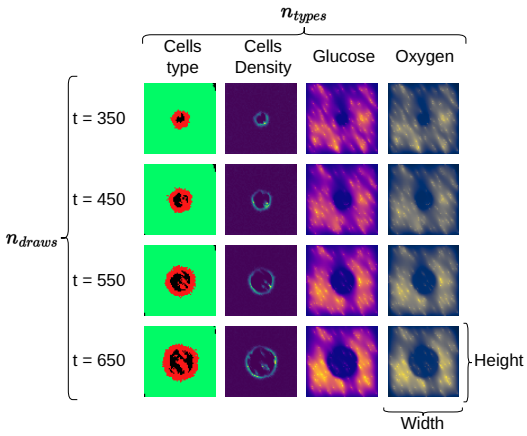

Figure 4.5: Set of matrices corresponding to one sample of the dataset

The simulation varies in time and several iterations of these matrices needs to be drawn from it. Hence, one sample of the dataset corresponds to a set of $n_{draws} \times n_{types}$ matrices (see Figure 4.5). n_{draws} is the number of times we draw matrices from the simulation and n_{types} is the number of different matrices we draw from the simulation at each time.

The program created for generating the dataset takes multiple settings as input:

- The size of the image (height and width) which has been set to 64×64 in order to have even-sized matrices when going through max-pooling operation in the CNN (see subsection 4.3.1) while not diverging too much from the original size of 50×50 in the paper of Moreau et al. [1].
- The first draw is generated after 350 hours of simulation corresponding to the start of the RT treatment.
- The following draws are generated every 100 hours.
- The number of timesteps n_{draws} is directly impacted by the interval between two timesteps. The simulation ends after 1150 hours, by the time the tumour is either cured or out of control. To reach this time with an interval of 100 hours, we need 8 timesteps.
- The parameters of interest and the range of them (see subsection 4.2.1).
- The number of samples in this dataset is a trade-off between accuracy and computation time. It can also heavily influence the overfitting of the NN. After some testing, we decide to use a dataset of 3200 samples.

4.2.1 Parameters

The simulation used to create each sample is initialised with specific values for parameters. These values are generated at random uniformly between a minimum and a maximum.

Cell cycle and nutrient consumption

The minimum and maximum values of the parameters have been defined as a percentage of the default value:

$$\min(param) = (1-\alpha)\ \text{default}(param) \quad \text{and} \quad \max(param) = (1+\alpha)\ \text{default}(param) \tag{4.9}$$

with default($param$) representing the default value of the parameter.

We re-introduce the condition between the critical level and the average consumption in subsection 4.1.3, and we consider Equation 4.4 that come out of it :

$$\begin{aligned} q_{critical} &> \mu_{cancer} \\ \lambda_{critical} \cdot t_{total} \cdot \mu_{healthy} &> \mu_{cancer} \end{aligned} \tag{4.10}$$

When considering the minimum and maximum of each parameter, the condition becomes:

$$\begin{aligned} \tfrac{3}{4} \cdot \min(t_{total}) \cdot \min(\mu_{healthy}) &> \max(\mu_{cancer}) \\ \tfrac{3}{4} \cdot (1-\alpha) t_{total} \cdot (1-\alpha)\mu_{healthy} &> (1+\alpha)\mu_{cancer} \\ \tfrac{3}{4} \cdot (1-\alpha)^2 t_{total} \cdot \mu_{healthy} &> (1+\alpha)\mu_{cancer} \end{aligned} \tag{4.11}$$

When replacing the value for the oxygen and the glucose:

$$\begin{cases} \frac{3}{4} \cdot (1-\alpha)^2 t_{total} \cdot \mu_{g,healthy} & > (1+\alpha)\mu_{g,cancer} \\ \frac{3}{4} \cdot (1-\alpha)^2 t_{total} \cdot \mu_{o,healthy} & > (1+\alpha)\mu_{o,cancer} \end{cases} \tag{4.12}$$

which yields a result of $\alpha < 0.63$ when solved. When rounded and applied to the parameters we obtain the range described in Table 4.5.

These ranges could be tweaked by introducing different α for each parameter and by constraining each α with the same conditions. In our case, we choose a unique α to keep a similar range scale with all the parameters.

Parameters	Default Value	Minimum	Maximum
Cell cycle	24	10	38
Average healthy glucose absorption	.36	0.144	0.576
Average cancer glucose absorption	.54	0.216	0.864
Average healthy oxygen consumption	20	8	32
Average cancer oxygen consumption	20	8	32

Table 4.5: Default value and range of the different parameters

4.2.2 Treatment planning generation

The behaviour of the tumour growth simulation is very important even without external RT treatments. However, the goal is to analyse the tumour reaction to proton therapy treatment. We will now extend the dataset by adding a process of treatment planning. Those three different datasets have been constructed:

- Tumour growth simulations without any external treatment. Cancer cells expand indefinitely.
- Tumour treatment with the baseline treatment planning consisting of one dose of $2Gy$ administered every 24 hours for 30 consecutive days. All treatments start after 350 hours of simulation to accurately represent a grown tumour at the start of the treatment.
- The final dataset contains tumour simulations treated with different treatments found by the best RL model from Martin et al. [2]. Figure 4.6 depicts the mean and standard deviation of these treatments along with the baseline for comparison. For the remainder of this work, we will refer to these treatments as the RL treatments.

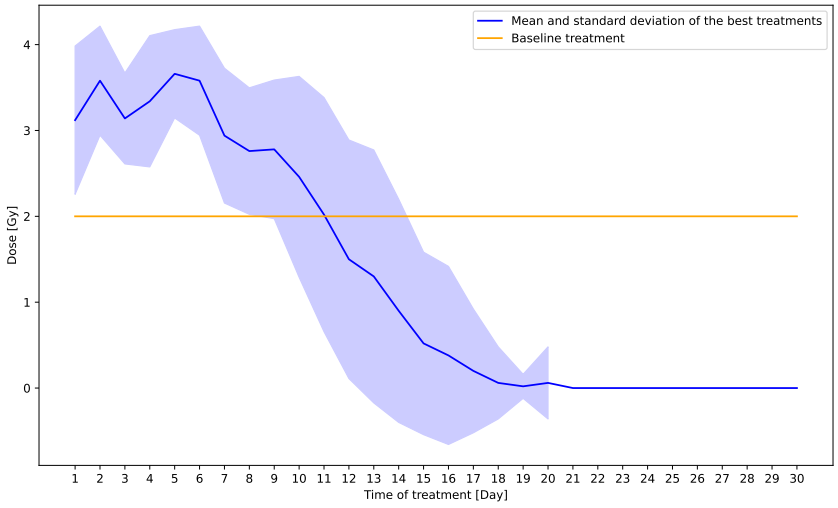

Figure 4.6: Dose fractionation for the baseline treatment and the treatment find by the RL model of Martin et al. [2]

In order to provide treatment information to our model, we added a fifth matrix type to the input of the NN. On each pixel of the new matrix, we store dynamically the cumulative dose of radiation received by this pixel from the beginning of the simulation.

4.3 Model

4.3.1 Architecture

The architecture proposed for this work is depicted on Figure 4.7. It is inspired by the work of Guo et al. [30] for the parameter identification of an elastic-plastic constitutive model and of Wu et al. [29] for video classification. It is highly modular with many possible settings.

The neural network takes as inputs the $n_{draws} \times n_{types}$ matrices with size $Height \times Width$. Each set of the four types of matrices goes through its own pipeline. Each pipeline is a CNN (subsection 2.2.4) to extract the information of the matrices. It is composed of alternating convolution and max-pooling layers followed by a fully connected linear layer to express the information at that time of the simulation. The next part is a collection of LSTM blocks linked together to handle the sequential vectors found by the CNN. We have an output vector for each block corresponding to the new hidden state (subsection 2.2.5) of the LSTM block. Finally, these vectors go through a fully connected linear layer with one hidden layer before the prediction of the outputs.

The neural network can be adjusted to the task with the following parameters:

- n_{draws} is explained in section 4.2 and controls the number of timesteps of the simulation used as input (4 on Figure 4.7).
- n_{types} is explained in section 4.2 and controls the type of image used as input (4 on Figure 4.7).
- $Height$ and $Width$ control the dimension of the matrices.
- n_{param} controls the number of parameters to predict.

Different components of the architecture can also be tuned to improve performance:

- n_{conv} controls the number of alternating layers of convolution and max-pooling (3 on Figure 4.7).
- n_{LSTM} controls the number of vertically stacked LSTM layers (1 on Figure 4.7). If this setting is zero, we remove entirely the LSTM part and $n_{in,LSTM} = n_{out,LSTM}$.
- $n_{in,LSTM}$ controls the size of the encoded vector that encapsulates the information of the matrices.
- $n_{out,LSTM}$ controls the size of the output of each LSTM blocks.
- $n_{feed\ forward}$ controls the number of neurons in the final hidden layer. This layer is entirely removed if the setting is set to zero.

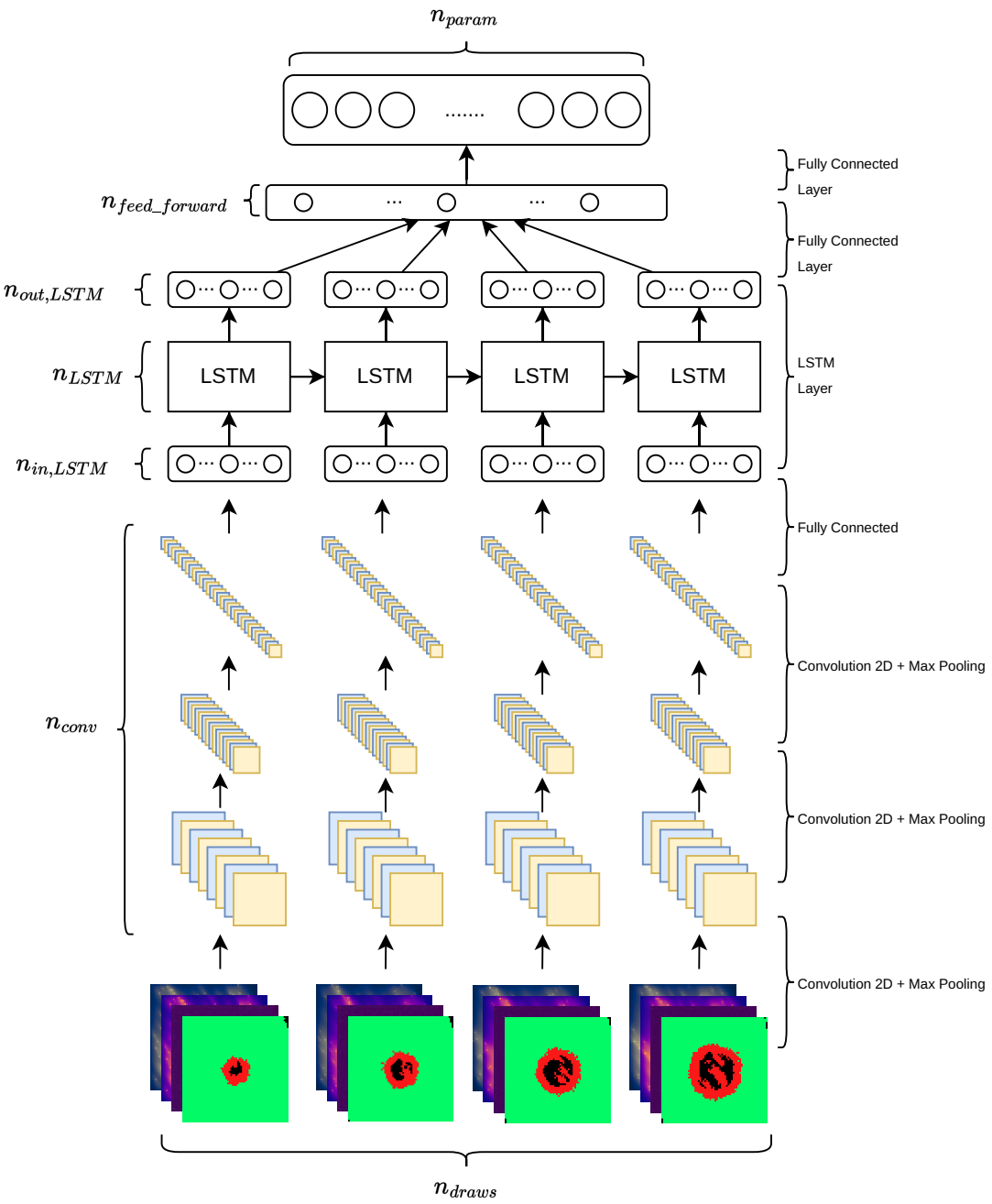

Figure 4.7: Architecture of the neural network

A significant part to mention is that the model does not predict directly the parameters. Instead, the final layer constrains the outputs between zero and one using the sigmoid function as activation function. During training, the parameters are normalised between zero and one, therefore the parameters used during training are expressed using the formulation:

$$y = \frac{(\xi - \xi_{min})}{(\xi_{max} - \xi_{min})} \tag{4.13}$$

This step is often done in parameter identification tasks to mitigate bias towards specific parameters as mentioned in section 3.3.

4.3.2 Training

The initial dataset is split into three parts: the training dataset, the validation dataset and the evaluation dataset with a split ratio from the original dataset of 64%, 16% and 20% respectively.

The training dataset is used to update the weight of the model using the gradient descent algorithm. The criterion used is the Mean Square Error (MSE):

$$MSE(\widehat{y}, y) = \frac{1}{P}\sum_{i=0}^{P}\left(\widehat{y}(\mathbf{x}_i) - y_i\right)^2 \tag{4.14}$$

It is one of the most popular and widely used loss function for a regression task. This criterion is measured for each epoch on the same training set but also on the validation set. It allows us to plot the learning versus validation curve to assess how well the model fits the training data but also how it is able to generalise on new, unseen inputs. Our model is trained on 300 epochs.

A common difficulty when training a model is overfitting. It happens when the parameters of the model fits too closely to the data, in that case the algorithm is not able to generalise on new data. Multiple techniques exist to avoid overfitting but two of them are implemented in this work: regularization and early-stopping.

Regularization is a method that aims at reducing the complexity of the model. It achieves this by introducing a new regularization term in the objective function that penalizes large weights. Two types of regularization exist:

- L1 regularization adds a penalty on the absolute value of the weight: $J(\mathbf{w}) = MSE\left(\widehat{y}(\mathbf{w}), y\right) + \lambda \sum_{w \in \mathbf{w}} |w|$
- L2 regularization adds a penalty on the squared value of the weight: $J(\mathbf{w}) = MSE\left(\widehat{y}(\mathbf{w}), y\right) + \lambda \sum_{w \in \mathbf{w}} w^2$

In our work, L2 regularization is used and λ is a hyperparameter.

Early stopping is also a method targeting the complexity of the model. It continuously observes the validation loss and interrupts the training if this loss is not improved during a specific number of epochs. The algorithm also allows a small margin of error in the loss. The number of epochs without loss improvements before stopping is called the patience and the small margin of error is the delta, they are considered as two hyperparameters that can be tweaked to improve performance.

Chapter 5

Analysis

5.1 Dataset exploration

This section aims to explore the dataset to improve our knowledge of the information it contains. It uses traditional methods like dimensionality reduction (subsection 2.2.1) or relevance between features and target (subsection 2.2.2) to uncover insights on our data. These analyses enable us to further improve some of our methodology when studying the inherent similarity (see section 5.2) and to obtain analysis pointer when looking at the result of our NN (see chapter 6).

5.1.1 Dimensionality reduction

To understand relationships between the features constituting our data, visualisation is an insightful approach to highlight complex interactions. In our case, we have a huge space of features with a size of maximum $n_{draws} \times n_{types} \times Height \times Width = 8 \times 4 \times 64 \times 64 = 131\ 072$ features. In order to help visualise this large amount of information, we used dimensionality reduction (see subsection 2.2.1) and specifically PCA to reduce the number of dimensions to two. Higher number of dimensions could encompass more information but at the cost of harder chart to analyse. We obtain Figure 5.1 after highlighting the value of the different parameters to predict.

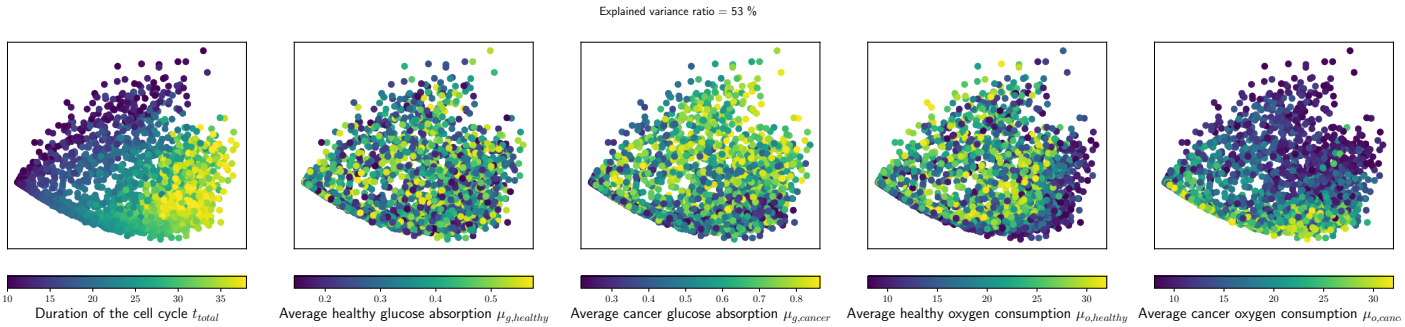

Figure 5.1: Dimensional reduction of the dataset without treatment using PCA. Parameters to predict are shown with the color gradient

From this chart alone, we can already affirm that the duration of the cell cycle t_{total} has a big impact on the overall simulation and therefore could be one of the easiest parameter to predict. We also observe that the simulation is far less structured around the other parameters. These observations are fairly logical due to the fact that the duration of the cell cycle has a direct impact on the simulation and is a driving force in the tumour evolution compared to the other parameters.

When looking at the chart, the other parameters can be ordered by their perceptible heterogeneity:

1. The average oxygen consumption of cancer cells $\mu_{o,cancer}$
2. The average glucose absorption of cancer cells $\mu_{g,cancer}$

3. The average oxygen consumption of healthy cells $\mu_{o,healthy}$
4. The average glucose absorption of healthy cells $\mu_{g,healthy}$

With the last 2 parameters having little or no visible influence.

The higher spread of samples of $\mu_{o,cancer}$ and $\mu_{g,cancer}$ can be explained by the fact that cancer is invasive and when the nutrient consumption change, cancer cells take advantage of the situation to reproduce much more whereas healthy cells enter quiescence when the density is too high. This may also be explained by the number of cells on the simulation given that cancer cells are present on more pixels on average.

Lastly, we ordered the consumptions of oxygen before the glucose because the chart shows a small but significant difference between the nutrients' consumption. This may be caused by the scaling of the oxygen supplies and consumptions compared to glucose.

The order of heterogeneity we observe can already give us indications about the possible performance of our model. The next sections go further into each parameter and account for the simulation time and the image type. PCA has been carried out for each different outputs of the simulation that we used for our prediction. The values of the parameters are again shown using a gradient of color.

Duration of the cell cycle (t_{total})

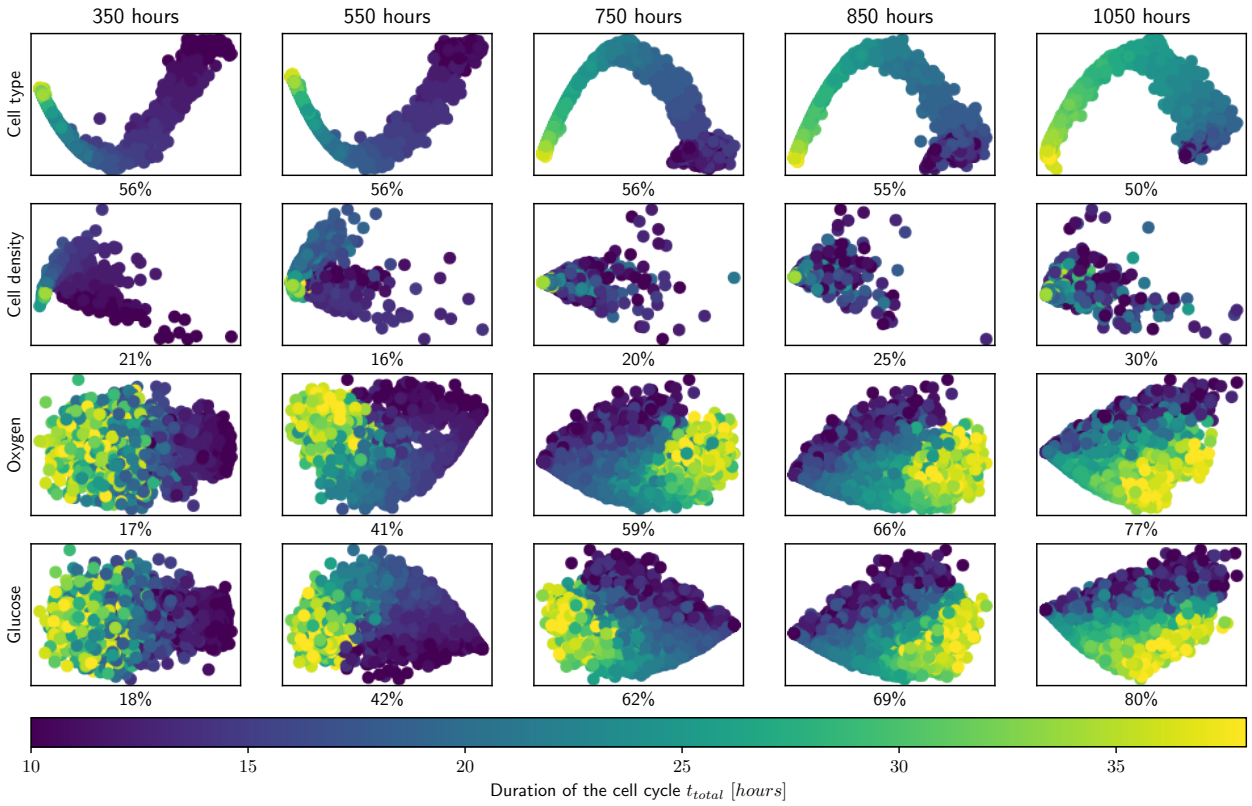

Figure 5.2: Dimensional reduction of the different images in the dataset without treatment highlighting the heterogeneity of the duration of the cell cycle. The explained variance ratio of the PCA is displayed below each graph

The striking thing to notice on Figure 5.2, and which is even more obvious when compared to the other parameters, is the strong heterogeneity of the duration of the cell cycle. We already noticed this when observing the dimensionality reduction of all the images at once.

By looking at each image type independently, other details stand out:

First, the image capturing the cell type is evolving during the simulation. Samples with a shorter duration of cell cycle seem to be more scattered at the start of the simulation while longer durations are all stacked together. On the contrary, the phenomenon is inverted through the simulation to reach the opposite observation at the end. In fact, simulations

with shorter cell cycle lead to very fast tumour growth at the start while longer durations correspond to barely developed tumour (see section 4.1.6). The consequence is that simulations with **shorter durations** express more their characteristics **at the start** and thus are more scattered. On the other hand, later in the simulation, tumours with faster cell cycle die of overpopulation, leading to a decrease in expressiveness. The opposite occurs with tumours with **longer cell cycle** reaching their peak development **at this end**.

We denote that this phenomenon can also be observed at a lesser scale in the three other types of images. We notice that the cell density points are very scattered and no real structure emerges. More complex patterns may be present but not observable due to the low explained variance ratio.

Finally, the last observation in this chart is the increase in the explained variance ratio over time for the glucose and oxygen matrices. This may be explained by the fact that when we observe these two outputs over time, we notice that the nutrients depletion zones formed by the nutrients consumptions of the tumour increase, leading to more information express through dimensionality reduction.

Average nutrients consumptions of cancers cells ($\mu_{g,cancer}$ and $\mu_{o,cancer}$)

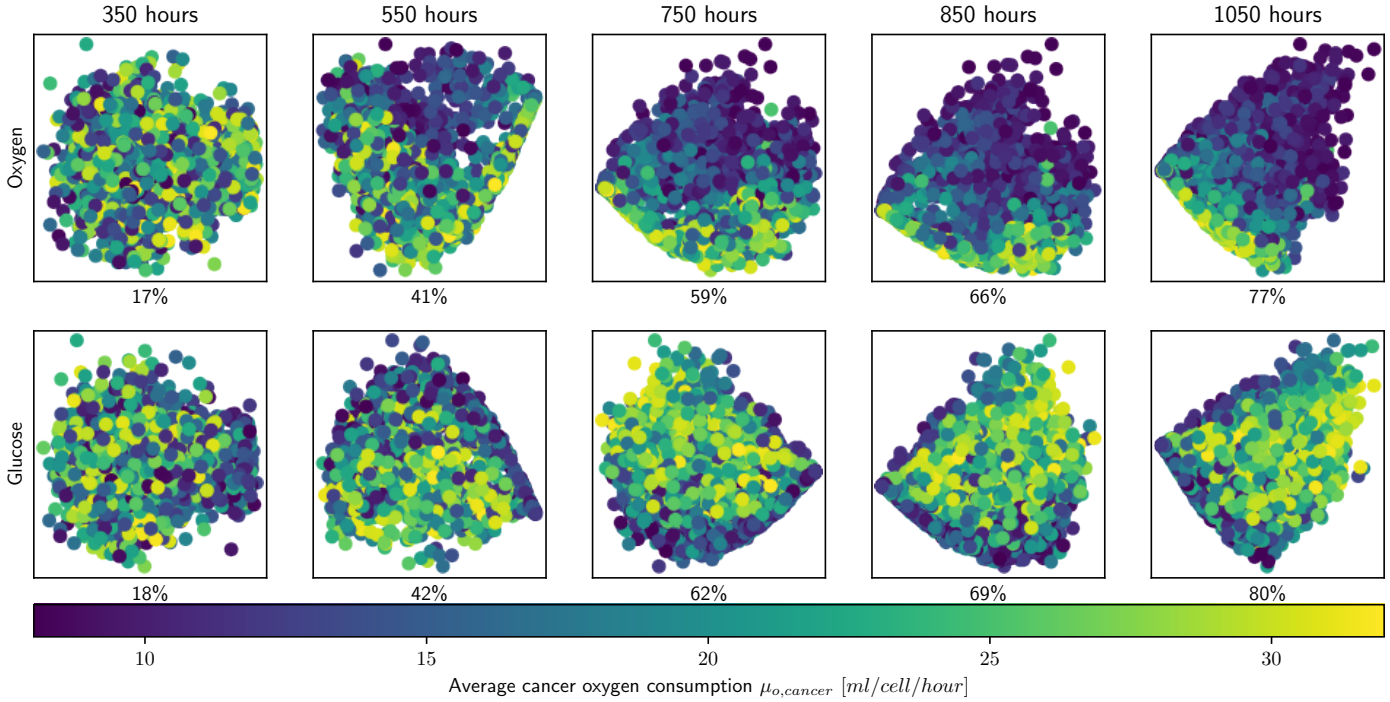

Figure 5.3: Dimensional reduction of the different images in the dataset without treatment highlighting the heterogeneity of the average oxygen consumption of cancer cells. The explained variance ratio of the PCA is displayed below each graph

The PCA graphs for these two parameters are similar except for the oxygen and glucose matrices that are swapped. For this reason, only the chart describing the oxygen consumption is shown (Figure 5.3).

In addition, charts with no perceptible information are not shown. The full images are available in annexes (section 9.1).

As already mentioned in subsection 5.1.1, these two parameters appear to have an overall lesser impact on the structure of the images. The nutrient's matrix corresponding to the parameter seems to start with no structure but shows an evolution over time towards a greater heterogeneity of the parameter, specifically on lower values.

We also notice a less perceptible but important observation. The dimensionality reduction of the nutrient's matrix opposite to the one we are studying (the glucose's matrix in this case) shows also a greater heterogeneity over time. The difference with the previous phenomenon is that this heterogeneity leans towards higher values of the parameter. This property is explained by another observations done while studying the impact of the values of the parameter on the simulation in section 4.1.6.

In terms of result, these two observations leads us to believe that the prediction of our NN on **both of these parameters should have a better prediction the more draws we have**.

Average nutrients consumptions of healthy cells ($\mu_{g,healthy}$ and $\mu_{o,healthy}$)

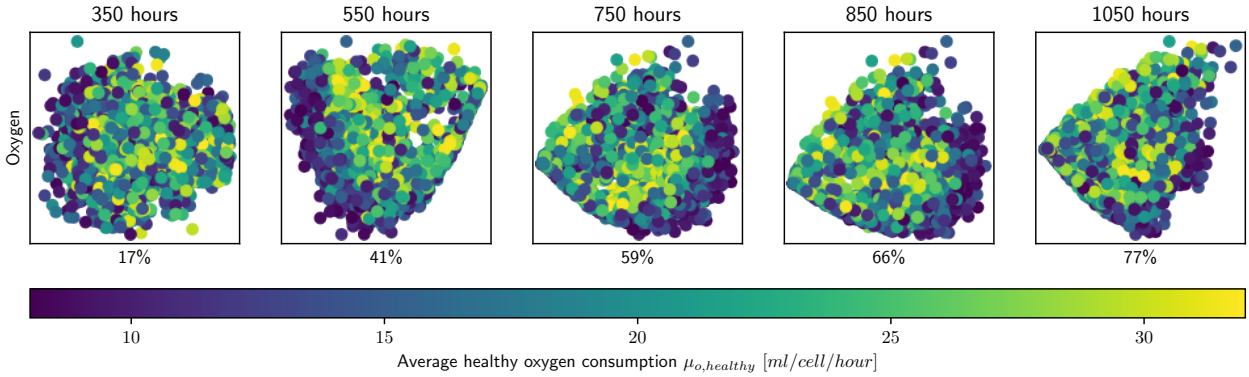

Figure 5.4: Dimensional reduction of the different images in the dataset without treatment highlighting the heterogeneity of the average oxygen consumption of healthy cells. The explained variance ratio of the PCA is displayed below each graph

As we have already done in the previous section, only the graph of oxygen consumption is shown and analysed due to the symmetry with regard to nutrients.

When observing the dimensional reduction while highlighting the average nutrients consumptions of healthy cells in subsection 5.1.1, we already notice little or no apparent structure around these parameters. Figure 5.4 allows us to situate this limited amount of structure in the nutrient's matrix corresponding to the parameter (the oxygen's matrix on the chart).

Impact of doses

The above analysis was done on a dataset with no treatment applied. We also carry out these analyses using the two datasets with treatment. Unfortunately, the reduced datasets show no distinctive sign of structure around the parameters. Consequently, we suspect a loss of performance while using treatment datasets.

Conclusion

All the information given above should be understood with care. Indeed, PCA, despite being a powerful algorithm, does not capture the entire dataset structure. In addition, PCA finds only new axis that are linear combination of the original one, non-linear structures can thus not be described properly by this algorithm. Non-linear dimensional reduction techniques have been tested (see section 9.2) and give us similar results to the one shown here.

Despite this information, the analyses give us insight about the repartition of the sample in the dataset. This repartition allows us to gain clues about the possible performance of our model. To summarise, the duration of the cell cycle t_{total} has a well-defined structure on the dataset, and we suspect a higher impact on the performance. This impact is exhibited on all the matrices which could also impact the inherent similarity analysis (see section 5.2). The two parameters describing the nutrients consumption of cancer cells have a lesser impact, but their influences increase with the number of draws. This means that performance of these parameters could improve over time. Finally, the two last parameters do not displayed any meaningful patterns in PCA that could impact the performance.

5.1.2 Mutual Information

Another interesting aspect of our dataset analysis is the relation between each feature and the target parameters. This is usually done using mutual information (subsection 2.2.2). In this section we analysed the results of the mutual information between each matrix and each target parameters. In practise, we consider each element of each matrix as a random variable X and the target parameter as another random variable Y. Each sample in the dataset is therefore an observation of these random variables. The mutual information $I(X, Y)$ is then computed for every element and normalised by the entropy of Y to obtain a value between 0 and 1 (subsection 2.2.2).

While losing information about the values of the parameters, this method allows gaining insight about the location of important features for prediction. This may be useful when looking at the weights of the CNN or the option of cropping the image in some places to decrease the input size. Mutual information may also be able to uncover non-linear patterns that PCA was not able to.

As already done in the previous part, the next sections will be divided by parameter. Specific observations may be redundant with the previous sections and will not be repeated.

Duration of the cell cycle (t_{total})

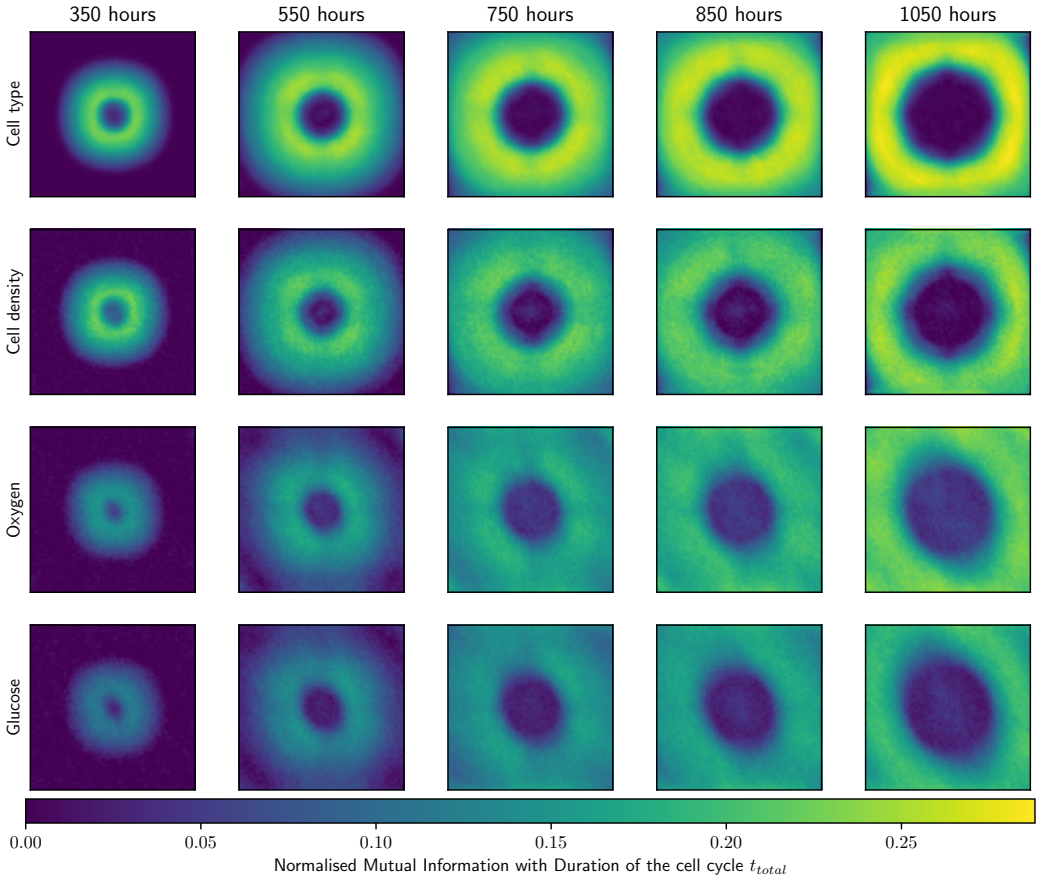

Figure 5.5: Normalised mutual information between matrices and the duration of the cell cycle on the dataset without treatment

When observing the MI for the cell cycle on Figure 5.5, the first striking result to observe is the amount of information between the cell density matrices and the duration of the cell cycle which is surprising since the dimensionality reduction did not depict such significant structure.

We also notice that the mutual information forms a halo around the centre of the simulation. It represents the region of growth of the tumour, the duration of the cell cycle will directly affect this zone and this shape is easily explained. The consequence of this observation is that all the pixels in the matrices, even ones at the border contains important information and are relevant to the task.

Finally, the main observation that is not shown in the dimensionality reduction section is the increase in mutual information over time for all the matrices. While small, it could imply an increase in the performance over time for this parameter.

Average nutrients consumptions of cancers cells ($\mu_{g,cancer}$ and $\mu_{o,cancer}$)

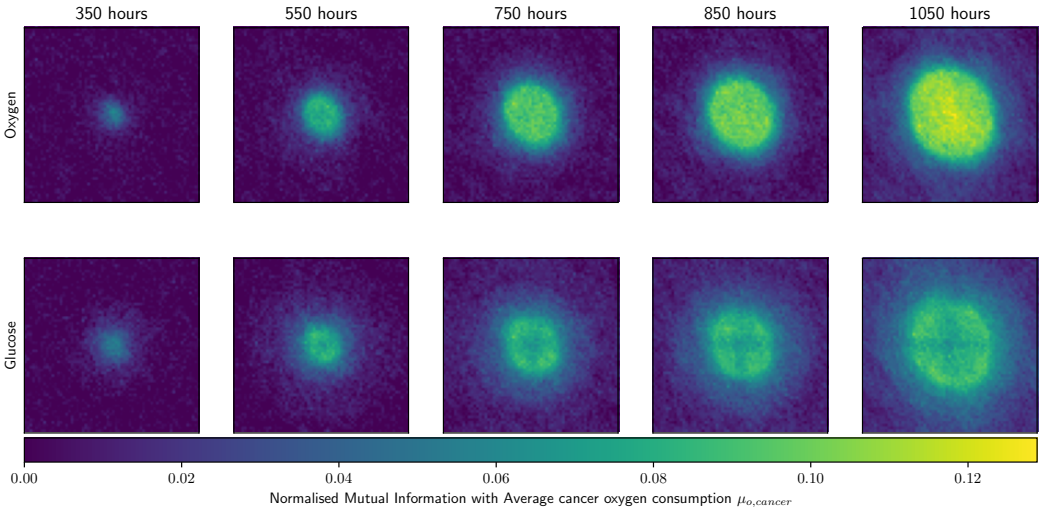

Figure 5.6: Normalised mutual information between matrices and the oxygen consumption of cancer cells on the dataset without treatment

Figure 5.6 and Figure 5.7 shows the consumption of both nutrients by cancer cells. The first thing to notice is that the information is not shaped as a halo like the previous parameter. Instead, it is regrouped at the centre in a round region that grows over time. This zone is where the nutrients depletion zones is created and where cancer cells die of overpopulation, consequently it contains the most of the information about changes in the nutrients' consumption.

These charts also allows us to confirm what is observed in the dimensionality reduction section. Indeed, the longer the simulation lasts the higher is the MI and on more pixels.

We also notice that the density matrix of the other nutrient than the one we are studying is also heavily impacted. This correlation between the nutrients is explained in section 4.1.6.

The mutual information between these two parameters and the other matrices is barely noticeable and as a result cannot produce any useful interpretation. The full images are nevertheless available in the annexe (see section 9.3).

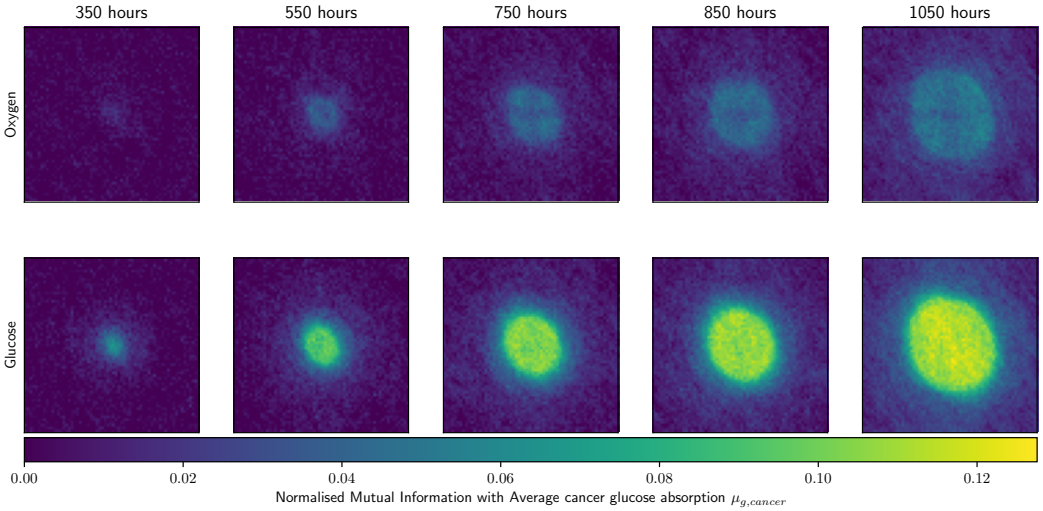

Figure 5.7: Normalised mutual information between matrices and the glucose absorption of cancer cells on the dataset without treatment

Average nutrients consumptions of healthy cells ($\mu_{g,healthy}$ and $\mu_{o,healthy}$)

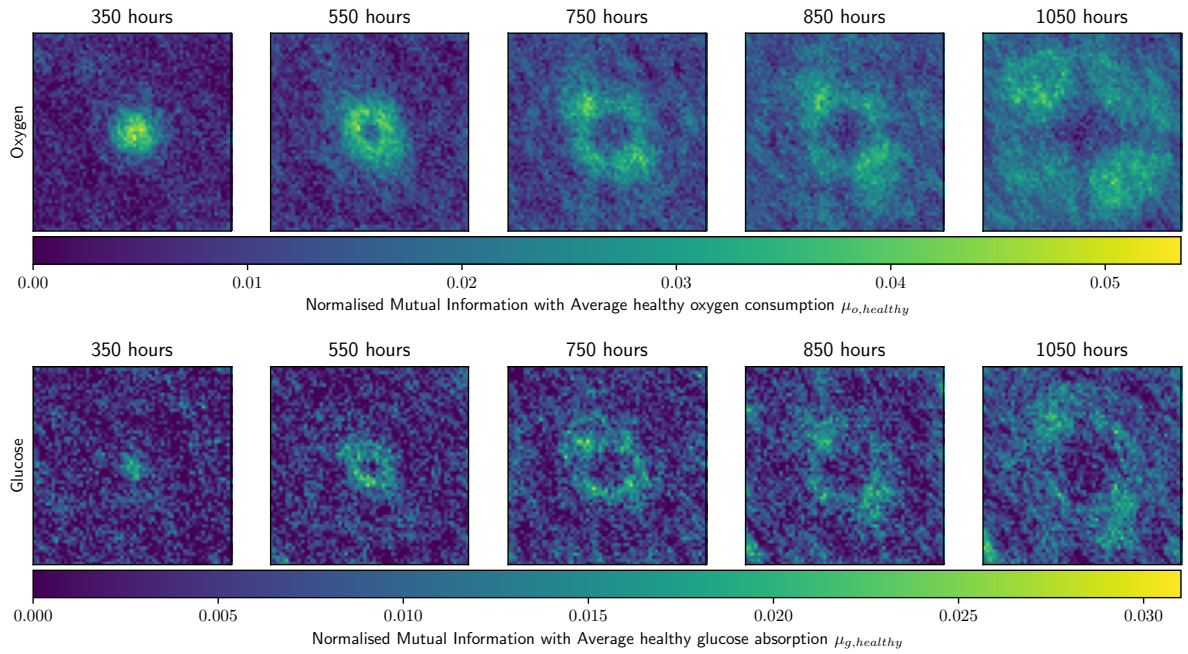

Figure 5.8: Normalised mutual information between matrices and both nutrients' consumption of healthy cells on the dataset without treatment

Figure 5.8 shows the consumption of both nutrients but for healthy cells this time. The amount of mutual information decreases compared to the other parameters. The mutual information is also much more random and scattered. This information is nearly essentially represented in the matrix related to the parameter we're studying. Though we cannot explain it, we notice an overall higher mutual information for the oxygen compared to the glucose.

Impact of doses

On the contrary of the dimensionality reduction, even when studying datasets with doses administered, we observe a significant amount of mutual information between parameters and matrices. However, the two treatments (baseline treatment or RL treatment) do not differ in their analysis. Figure 5.9 is displayed here with the duration of the cell cycle to demonstrate the effect of doses on the mutual information analysis for all parameter, but the observations we made in this subsection is true for all the parameters. The others charts are available in the annexe (section 9.3).

We observe a decrease in mutual information compared to the chart without treatment, this decrease is more noticeable the longer the simulation lasts. For that matter, the first row of the charts with and without treatment are often equivalent. This could result in stagnating performances while predicting the parameters with a NN not able to add information during the simulation.

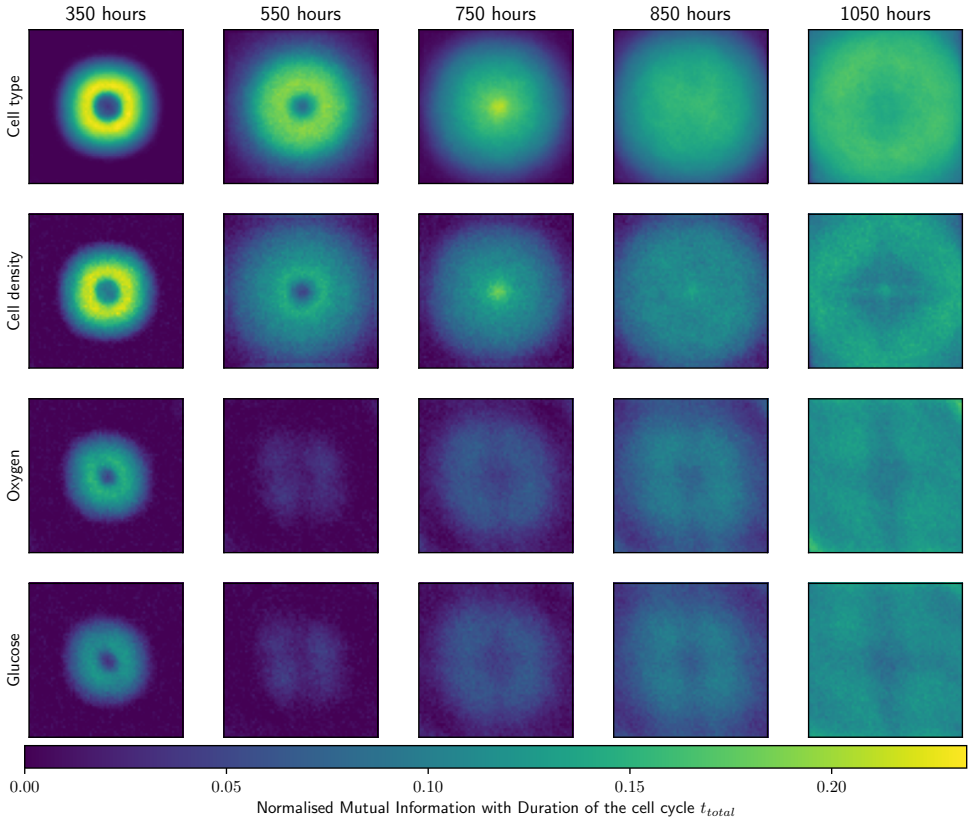

Figure 5.9: Normalised mutual information between matrices and the duration of the cell cycle on the dataset with treatment

Conclusion

The analysis done in this section brings us with new information about the spatiality of the parameters influences on the matrices. It also uncovers new insights not present in this dimensionality reduction, specifically when studying the impact of the RT doses.

To summarise, we observe a high relationship between the localisation of high mutual information values and the localisation of the tumour. While the duration of cell cycle t_{total} affects more the edge of the tumour where it grows, the nutrients' consumptions of cancer cells affect the centre of the tumour in which the overpopulation phenomenon takes place. Consequently, no part of the matrices could be cropped.

This section also allows us to come back to the ordering of parameters we made in subsection 5.1.1 but with tangible values:

1. The duration of the cell cycle t_{total} with a maximal mutual information of about 25%.
2. The average oxygen consumption of cancer cells $\mu_{o,cancer}$ with a maximal mutual information of about 12%.
3. The average glucose absorption of cancer cells $\mu_{g,cancer}$ with a maximal mutual information of about 12%.
4. The average oxygen consumption of healthy cells $\mu_{o,healthy}$ with a maximal mutual information of about 5%.
5. The average glucose absorption of healthy cells $\mu_{g,healthy}$ with a maximal mutual information of about 3%.

Even if the normalised MI is not very high, this ordering could possibly indicate the relative performance between parameters when observing the result of our model in chapter 6.

Finally, this section also confirms our intuition discussed in subsection 5.1.1, especially on the evolution in time of the influence of $\mu_{g,cancer}$ and $\mu_{o,cancer}$ on the matrices. This analysis also allows to claim that treatments doses reduce the mutual information on the matrices drawn from latter in the simulation. It means that the treatment could possibly have a negative impact on the performance of our model.

5.2 Inherent similarity between simulations

The simulation used in this work and described in section 4.1 is very stochastic and lots of processes occur randomly in it. Notably:

- The computation of the efficiency value depends on a random factor following a normal distribution (see subsection 4.1.2).
- The cell survival when irradiating depends on a probability and the repair counter is also incremented by a random value (see section 4.1.5).
- The initialisation of the nutrients sources locations is randomised, and their movements is defined by a random walk (see subsection 4.1.4).
- The location and stage of initial cells is also determined at random.

As mentioned in section 3.3, this stochastic property prevents us from being able to define the simulation as a function. Despite this information, we may still be able to estimate its reciprocal to a certain degree of error. This section aims to characterise the inherent similarity between different simulations and to express the prediction error of our model as an error between simulations.

In practise, we studied how different the output matrices of the simulation are between two independent runs separated by a certain difference in the initial parameters. We started by looking at the element-wise absolute difference between two matrices and while this absolute difference allows us to gain insight on the localisation and the repartition of the inherent difference in the simulation, it does not allow us to perform simple, automated comparison between matrices. For this reason, we need to select a metric able to provide us with a measure of similarity as a simple scalar value.

5.2.1 Selecting the optimal metrics

Many metrics have been defined in section 2.3, and we need to select the optimal ones for easier comparison between matrices. This process involves looking at the evolution of each metric when varying the difference in parameters and settle on those with the most expected behaviour.

We want to point out that each comparison is performed on 100 000 pairs of matrices sampled from our dataset with RL treatment (see subsection 4.2.2). We also choose to carry out this analysis on matrices after 550 hours because the treatment starts to get rid of the tumour at this time. The whole process is done two times and two metrics are selected independently: one for the continuous matrices and one for the discrete matrices.

Optimal metric for continuous matrices

Continuous matrices encompass the cell density matrix and both nutrients matrices. Among these matrices, we choose the one which we find is the most affected by changes in initial parameters: the oxygen matrix (the glucose matrix would have been a good choice too). This matrix helps us to determine the best metric. With a similar reasoning (see section 5.1), we justify to vary the cell cycle duration t_{total}.

After comparing the matrices with different metrics, we obtain Figure 5.10. The various metrics used measures either a similarity or a dissimilarity, and each does so at a different scale. In order to give a chance to all the metrics, we normalise the means between 1 (perfect similarity) and 0 (perfect dissimilarity). The standard deviation is also normalised to achieve same scale for all the metrics.

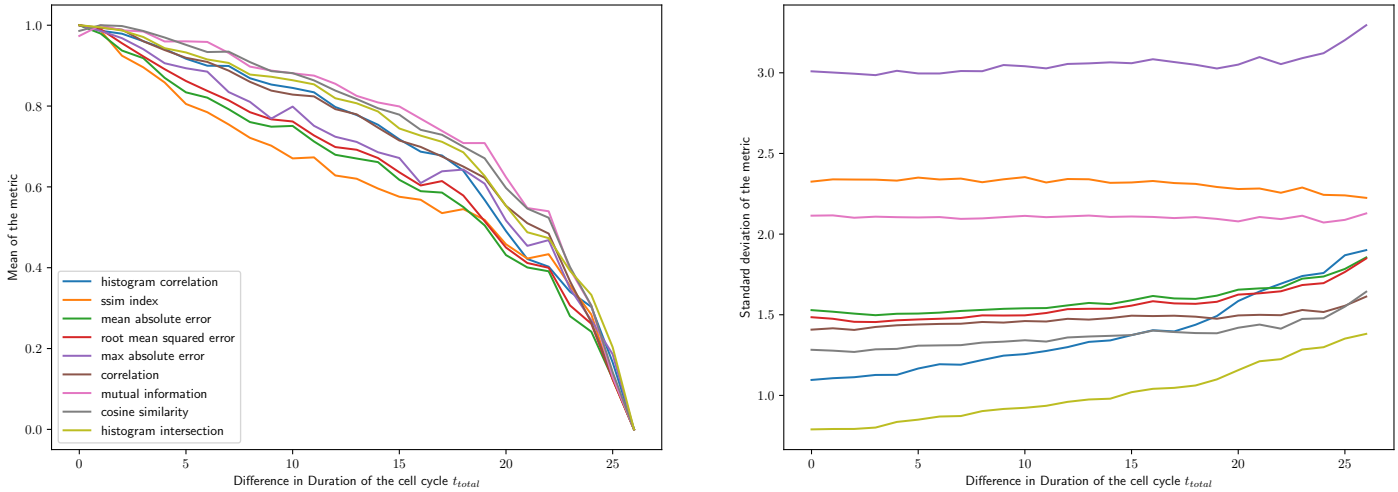

Figure 5.10: Evolution of the mean (left) and standard deviation(right) of different similarities between oxygen matrix in function of the difference in the duration of the cell cycle

Finally, to choose a metric among all, we define three scores representing property that our metric should possess:

1. First, we want our metric to be equal to 1 when there is no difference in the initial parameter and 0 when there is a maximal difference. This is summarises by asking our metric to be decreasing with respect to t_{total}. We also add that this relation should be monotonically decreasing for consistency. It is formulated as:

$$\min_{\mathbf{x}} \left(\sum_{x_i \in \mathbf{x} | x_i - x_{i-1} > 0} (x_i - x_{i-1}) \right) \text{ with } \mathbf{x} \text{ a vector containing the different means.} \tag{5.1}$$

2. Secondly, we want the standard deviation to be as low as possible to represent a more stable and consistent metric:

$$\min_{\sigma} \frac{1}{N} \sum \sigma_i \text{ with } \mathbf{x} \text{ a vector containing the different standard deviation} \tag{5.2}$$

3. Finally, we authorise the standard deviation of our metric to increase with the difference of initial parameter to represent divergence between distant simulations:

$$\max_{\sigma} (\sigma_N - \sigma_0) \tag{5.3}$$

Table 5.1 shows the results of the three scores for the various metrics. We decide to pick the histogram intersection metric to assess the similarity between matrices with continuous values.

Metric	Monotonically Decrease	Low Standard Deviation	Standard Deviation Increase
Histogram intersection	0.0	1.014	0.592
Cosine similarity	0.015	1.375	0.36
Mutual information	0.027	2.104	0.014
Correlation	0.0	1.475	0.205
Max absolute error	0.077	3.056	0.286
Root mean squared error	0.011	1.56	0.364
Mean absolute error	0.002	1.595	0.327
SSIM index	0.023	2.312	-0.101
Histogram correlation	0.0	1.385	0.805

Table 5.1: Computed scores for the different metrics on continuous matrices

Optimal metric for discrete matrices

Discrete image includes only the cell type matrix. The analysis of the optimal metric is done separately of the other because each pixel can contain only three specific values, this may impact how a metric gather the similarity of the two matrices into a scalar value. We choose to study the cell type matrix under variation of the duration of the cell cycle t_{total} because we find this parameter to have more influence on this matrix (see section 5.1).

The method to find the optimal metric is the same as the one for continuous matrices except for the two new metrics specific to discrete image that are also added. The mean and standard deviation as a function of the difference of the cell cycle duration is depicted on Figure 5.11.

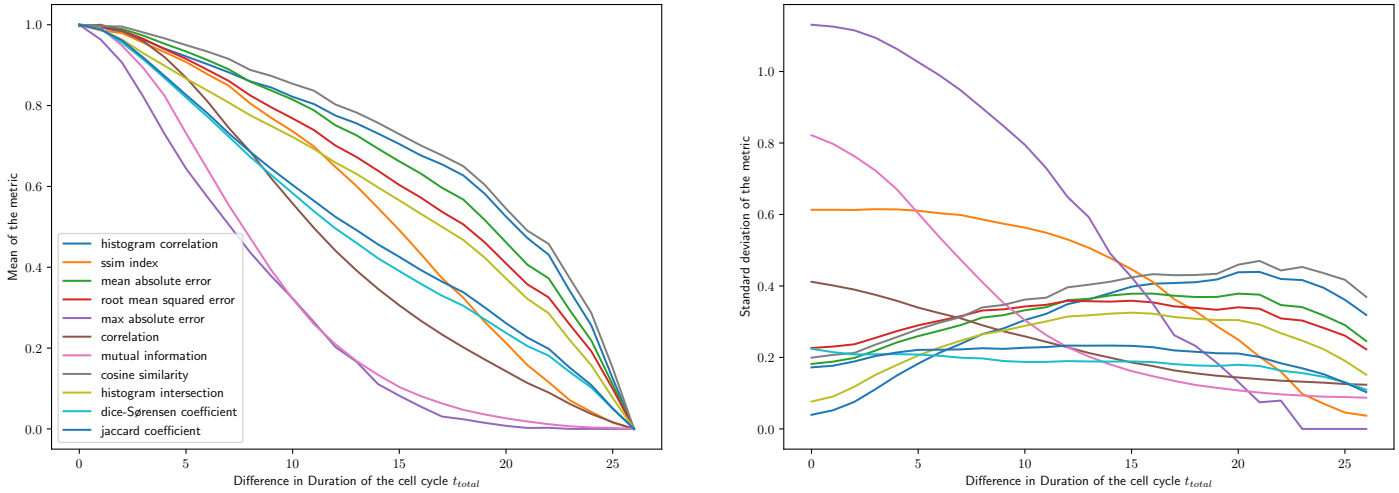

Figure 5.11: Evolution of the mean(left) and standard deviation(right) of different similarities between the cell type matrix in function of the difference in the duration of the cell cycle

Table 5.2 depicts the measure of the different score for each metric. The Jaccard coefficient, which is more adapted to discrete matrices, has been selected for his low standard deviation and nearly constant relation to the difference of initial parameter.

Metric	Monotonically Decrease	Low Standard Deviation	Standard Deviation Increase
Jaccard coefficient	0.0	0.203	-0.069
Dice-Sørensen coefficient	0.0	0.184	-0.114
Histogram intersection	0.0	0.246	0.075
Cosine similarity	0.0	0.364	0.17
Mutual information	0.0	0.321	-0.734
Correlation	0.004	0.236	-0.288
Max absolute error	0.0	0.564	-1.131
Root mean squared error	0.0	0.309	-0.004
Mean absolute error	0.0	0.312	0.064
SSIM index	0.0	0.421	-0.576
Histogram correlation	0.0	0.302	0.279

Table 5.2: Computed scores for the different metrics on discrete matrices

5.2.2 Automated estimation of the error between simulations

The results of our models are defined as the absolute difference between the true and the predicted values of the parameters. This subsection aims to define how this difference in parameters is reflected in the matrices and how well the NN performed in terms of simulation outputs. It enables us to compare the similarity between matrices of simulations differentiated by the error in prediction of the parameters. This method is described below:

For a given parameter, a matrix type and a number of draws:

1. Using the appropriate metric (subsection 5.2.1), we compute the similarity between the matrices as a function of the difference between parameter values.
2. We normalise this similarity between 0 and 1.
3. The function obtained allows us to determine the similarity between matrices based on the error in prediction.
4. Finally, we subtract the similarity from one to derive the error between matrices.

It occurs that the metric does not vary or increase with respect to the difference between parameter values. In those cases, the error between matrices is set to 0 to avoid negative values.

This process is conducted on the performance of our models and is depicted in chapter 6.

Chapter 6

Results

This chapter presents the performance of our NN (see subsection 4.3.1) on the prediction of certain parameters of the simulation. The optimal hyperparameters of our models are found through a search process that randomly selects 40 sets of hyperparameters and assess their performances. This process is carried out each time for a distinct dataset and number of draws. The optimal hyperparameters for each model are displayed in the annexes (see section 9.4) along with the overall influence of these hyperparameters (see section 9.5).

6.1 Duration of the cell cycle t_{total}

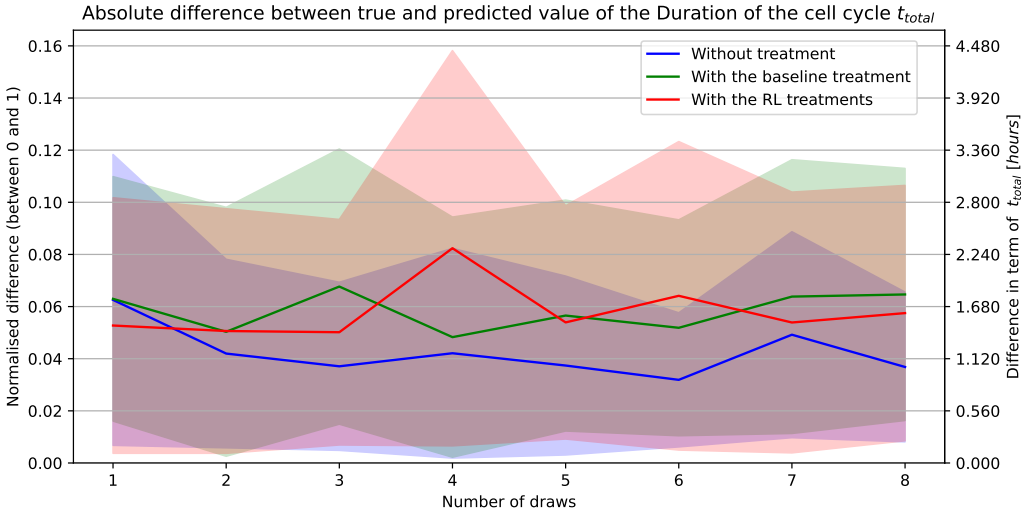

Figure 6.1: Mean and standard deviation of the absolute difference in prediction for the duration of the cell cycle t_{total}. Different scales are used on the image: the normalised error between 0 and 1 on the left, and the non-normalised error between the parameter range (on the right).

The duration of the cell cycle t_{total} is the parameter demonstrating a higher heterogeneity in subsection 5.1.1 and a larger mutual information in subsection 2.2.2. These analyses foreshadowed the performance seen on Figure 6.1. Indeed, the error for the three datasets is significantly lower compared to the other parameters. On the dataset without treatment, we also notice a slight decrease in error the longer the simulation lasts. This decrease does not appear when a treatment is applied. This phenomenon is one of the main observation in the dataset exploration section (see section 5.1).

Figure 6.2 allows us to gain information on how the performance of our model is translated to the four types of matrices. As observed when studying the dataset, the cell type is the matrix type most affected by the duration of the cell cycle, and consequently we notice a higher dissimilarity on the chart for this matrix. We also comment on the fact that the error between matrices for the dataset without treatment increase in relation to the number of draws even though the absolute

difference in prediction is decreasing. Our interpretation is that as the tumour grows without control, the simulations diverge extremely even for similar values of t_{total}.

To end the section about the duration of the cell cycle, we find that its prediction results are quite promising with an error of around 1.5 hours. We also observe that the model has the capacity to distinguish between two matrices with only 8% of dissimilarity in the worst case.

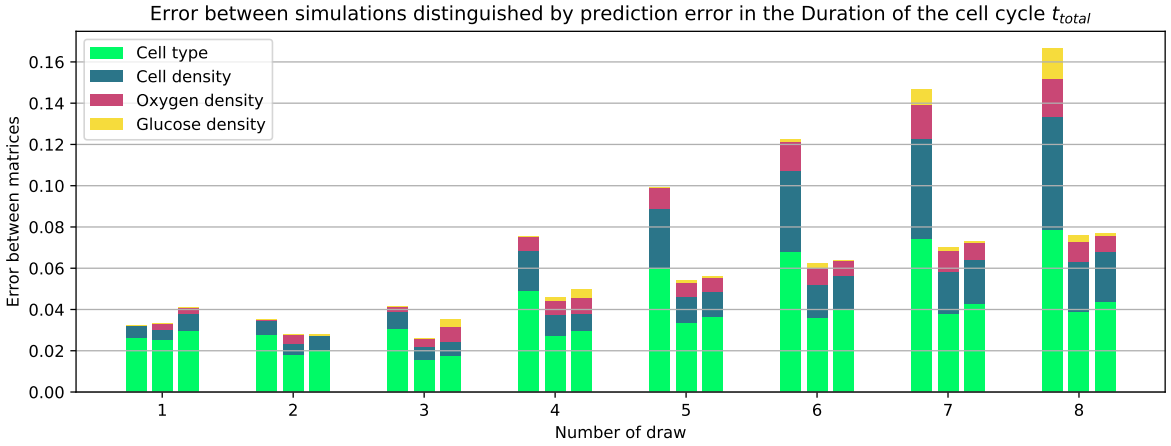

Figure 6.2: Error between simulations distinguished by the prediction error of the duration of the cell cycle t_{total} for the three datasets: without treatment (left bar), with baseline treatment (middle bar), and RL treatment (right bar). The method to calculate this error is described in subsection 5.2.2

.

6.2 Average oxygen consumptions of cancer cells $\mu_{o,cancer}$

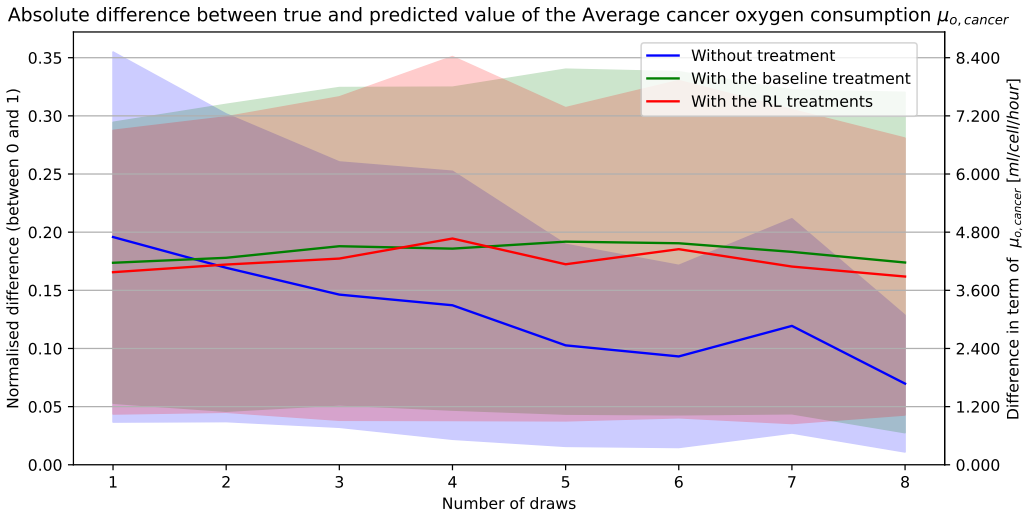

Figure 6.3: Mean and standard deviation of the absolute difference in prediction for the average consumption of oxygen by cancer cells $\mu_{o,cancer}$. Different scales are used on the image: the normalised error between 0 and 1 on the left, and the non-normalised error between the parameter range (on the right).

According to section 5.1, $\mu_{o,cancer}$ even if it is following t_{total} in terms of expected performance, a significant gap exists between t_{total} and the other parameters. As anticipated, this gap is observed in Figure 6.3 with an error in prediction of around 17% for datasets with treatments. However, on the dataset without treatment, we notice the same phenomenon as in the previous section with a decrease of the error in relation to the number of draws. Precisely, in comparison with t_{total},

this observation is significantly more expressed and allows the model to achieve a performance of similar scale to t_{total}. On the contrary, datasets with treatments are less dependent on the number of draws, and the performance is stagnating.

Figure 6.4 depicts the performance on the matrices of the simulation. As expected, we notice that the oxygen matrix is the one showing the highest dissimilarity due to its high MI with the target $\mu_{o,cancer}$. As observed in subsection 5.1.2, the glucose matrix is also highly correlated to this parameter. Consequently, the chart depicts a higher error on this matrix when predicting the average cancer oxygen consumption.

Finally, for the four average nutrient consumptions, we want to point out the fact that the similarities between matrices are not necessarily decreasing as a function of the difference in parameter values, leading to a lack of results in the error between simulations. For $\mu_{o,cancer}$, this phenomenon is mostly visible on the dataset with RL treatments.

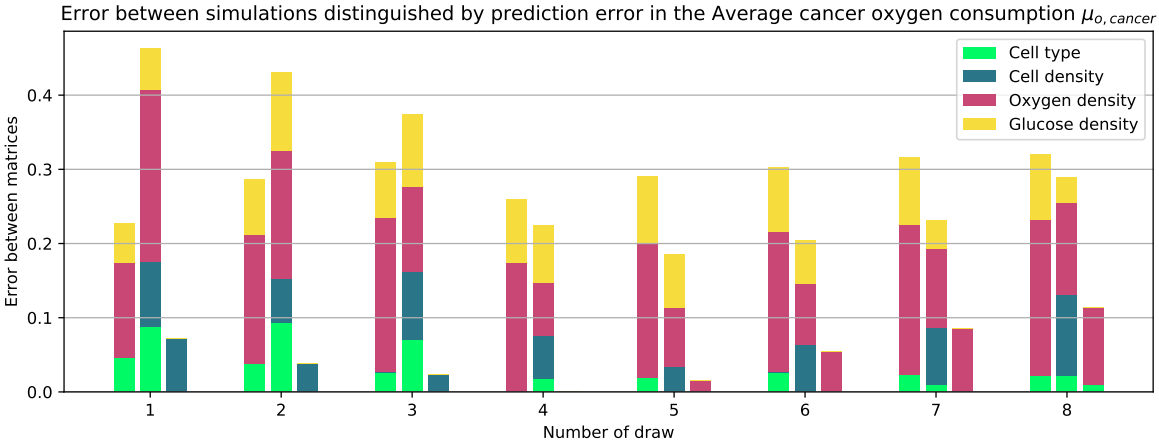

Figure 6.4: Error between simulations distinguished by the prediction error of the average oxygen consumption of cancer cells $\mu_{o,cancer}$ for the three datasets: without treatment (left bar), with baseline treatment (middle bar), and RL treatment (right bar). The method to calculate this error is described in subsection 5.2.2

.

6.3 Average glucose absorptions of cancer cells $\mu_{g,cancer}$

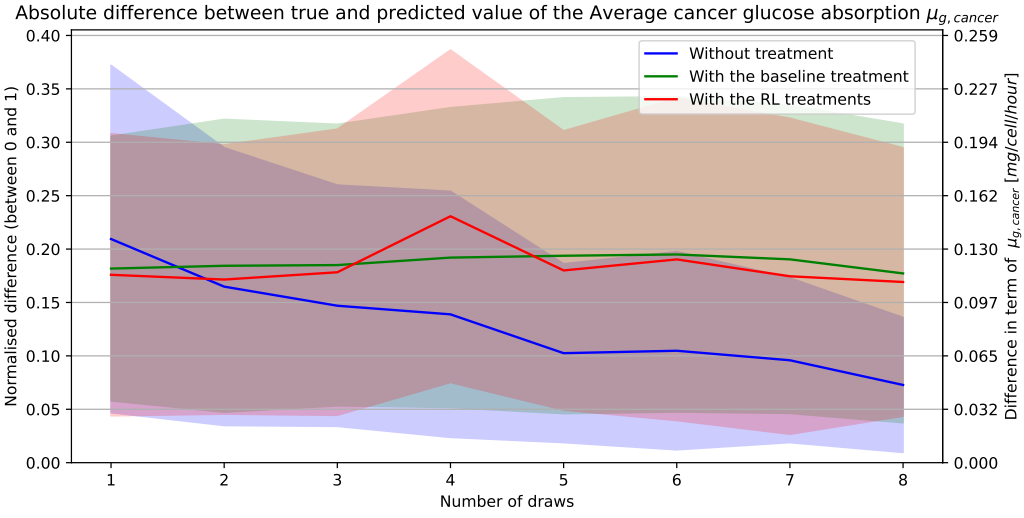

Figure 6.5: Mean and standard deviation of the absolute difference in prediction for the average absorption of glucose by cancer cells $\mu_{g,cancer}$. Different scales are used on the image: the normalised error between 0 and 1 on the left, and the non-normalised error between the parameter range (on the right).

The performance on the prediction of the average glucose absorptions of cancer cells $\mu_{g,cancer}$ is similar to its oxygen counterpart. Despite this, we observe a slight drop in performance of around 1% in Figure 6.5. It could be explained by the randomised hyperparameters searching algorithm, but we also observe a slight difference in heterogeneity on the dataset (see subsection 5.1.1).

Figure 6.6 depicts the dissimilarity between matrices corresponding to the prediction error. We observe a similar pattern as in the previous section but with the main dissimilarity coming from the glucose density matrix.

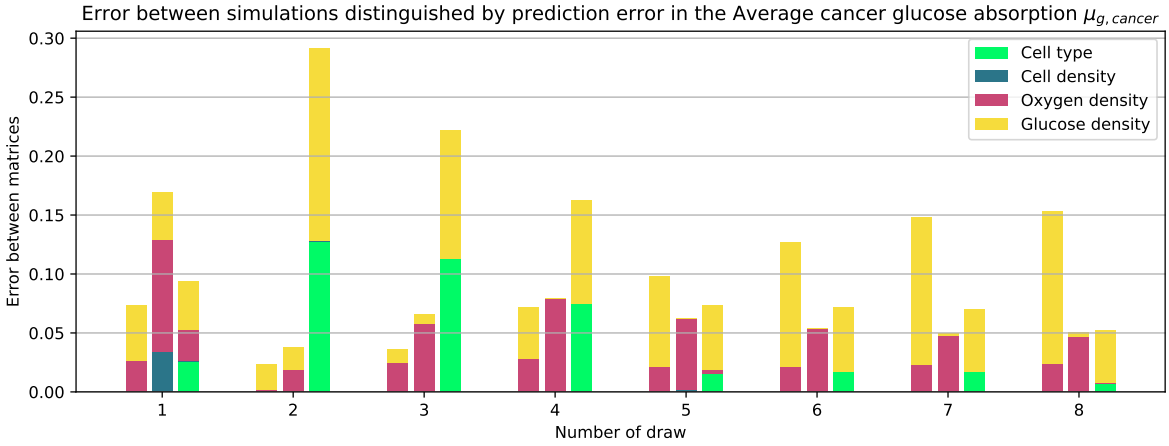

Figure 6.6: Error between simulations distinguished by the prediction error of the average glucose absorption of cancer cells $\mu_{o,cancer}$ for the three datasets: without treatment (left bar), with baseline treatment (middle bar), and RL treatment (right bar). The method to calculate this error is described in subsection 5.2.2

.

6.4 Average oxygen consumptions of healthy cells $\mu_{o,healthy}$

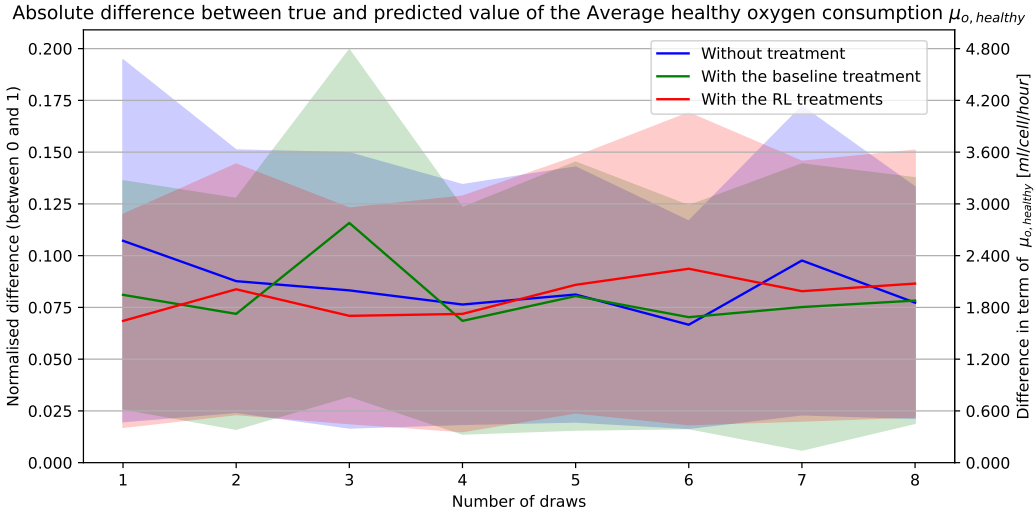

Figure 6.7: Mean and standard deviation of the absolute difference in prediction for the average consumption of oxygen by healthy cells $\mu_{o,cancer}$. Different scales are used on the image: the normalised error between 0 and 1 on the left, and the non-normalised error between the parameter range (on the right).

The performances of $\mu_{o,healthy}$ depicted in Figure 6.7 are the most surprising ones with a mean error of prediction of about $8-9\%$. While this result is promising for this work, no significant interpretation is found to explain this sudden increased performance.

Figure 6.8 shows that the prediction error is reflected on only the oxygen matrix, as it is the only one affected by this parameter (see subsection 5.1.2). Indeed, we observe that it is the only matrix type that possesses a low error. On this chart, we also observe the same phenomenon described in section 6.2 with no meaningful result available for the dataset with baseline treatment.

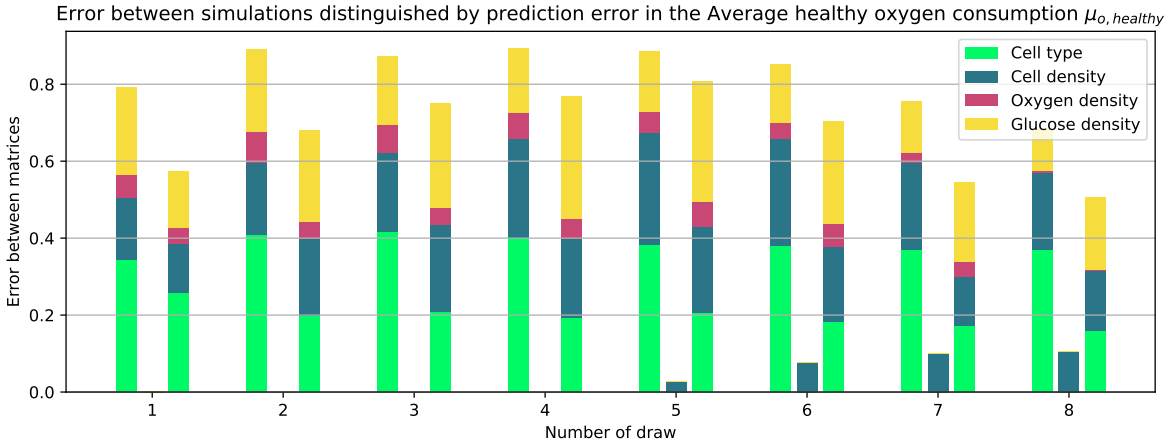

Figure 6.8: Error between simulations distinguished by the prediction error of the average oxygen consumption of healthy cells $\mu_{o,healthy}$ for the three datasets: without treatment (left bar), with baseline treatment (middle bar), and RL treatment (right bar). The method to calculate this error is described in subsection 5.2.2

.

6.5 Average glucose absorptions of healthy cells $\mu_{g,healthy}$

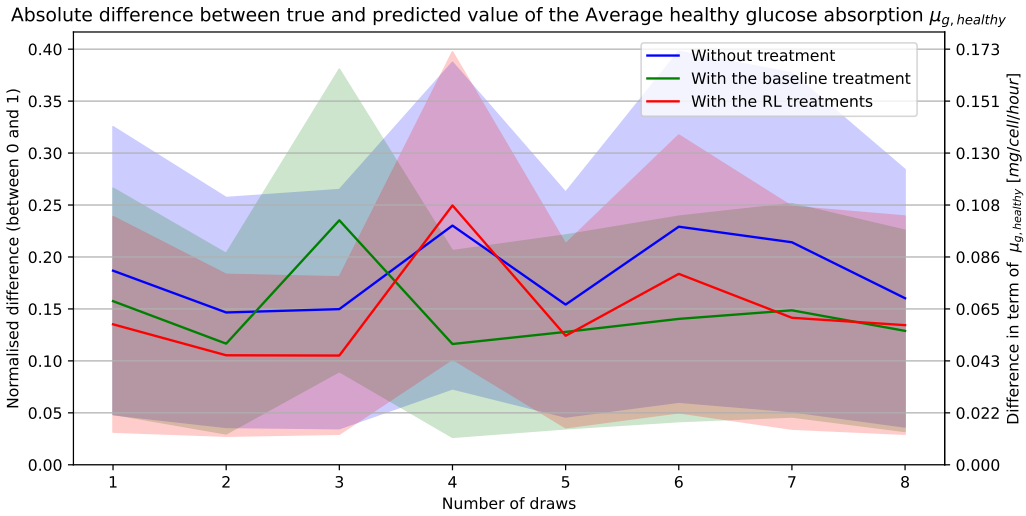

Figure 6.9: Mean and standard deviation of the absolute difference in prediction for the average absorption of glucose by healthy cells $\mu_{g,cancer}$. Different scales are used on the image: the normalised error between 0 and 1 on the left, and the non-normalised error between the parameter range (on the right).

As already observed for their counterparts on cancer cells, the average nutrient consumptions of healthy cells are similar in their behaviour (see Figure 6.9). However, compared to $\mu_{o,healthy}$, the prediction of parameter does not observe a surprising drop in prediction error. Consequently, the mean error in prediction is around 15%. One observation to notice that is already present but less perceptible in the previous section is the relative performance between datasets. We note that the dataset without treatment possesses lower performance compared to both datasets with treatments. Our interpretation is that without treatment, the cancer grows at the expense of healthy cells, and with less healthy cells,

$\mu_{g,healthy}$ and $\mu_{o,healthy}$ lower their influence on the dataset, leading to a higher prediction error. The error in simulation is shown in Figure 6.10. We notice that it is mainly expressed through the dataset without treatment and especially on the cell type matrix which possesses little or not MI with this parameter.

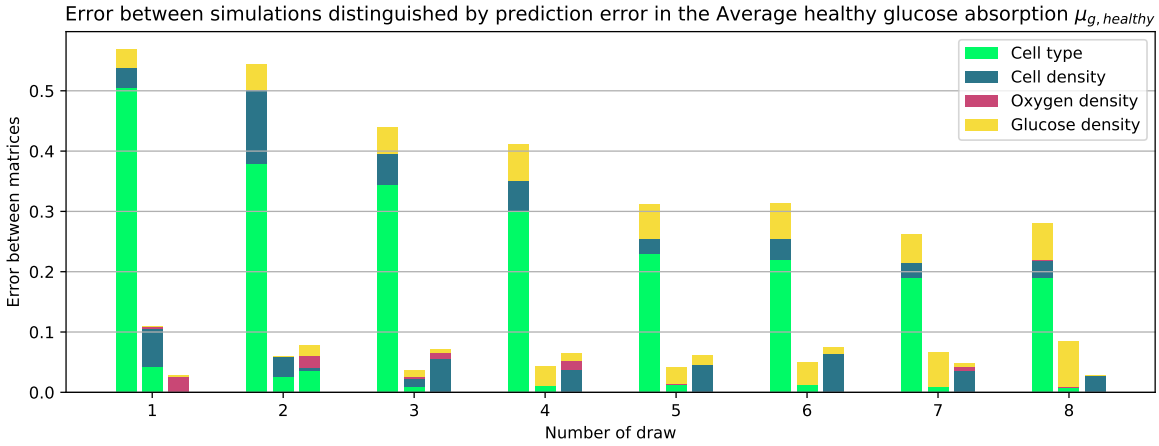

Figure 6.10: Error between simulations distinguished by the prediction error of the average glucose absorption of healthy cells $\mu_{g,healthy}$ for the three datasets: without treatment (left bar), with baseline treatment (middle bar), and RL treatment (right bar). The method to calculate this error is described in subsection 5.2.2

.

Chapter 7

Improvements and future works

This section is dedicated to explore possible improvements and potentially useful extension of the different methods used in the master's thesis.

- Among all the simulation parameters, only five parameters have been considered and implemented into the NN. In order to adapt the simulation efficiently to any situation, other parameters could be included in the prediction process. Increasing the number of parameters could allow the model to perform in a wider variety of contexts and possibly patients. The addition of parameters should always be supported by a control on the robustness of the architecture to a higher number of parameters. Finally, the parameters to add should be selected carefully because certain parameters are not designed to vary in the simulation. We advise starting studying the following parameters for prediction:
  - The radiosensitivity values: γ_{G_1},γ_S,γ_{G_2},γ_M,γ_{G_0}. Specifically when observing datasets with RT treatment.
  - The quiescent multiplier λ_{G_0} and the critical multiplier $\lambda_{critical}$ present in subsection 4.1.3 are respectively equal to 2 and $\frac{3}{4}$, but could vary and influence heavily the simulation.
- The method used to calculate the ranges of these parameters in subsection 4.2.1 could also be discussed. A single percentage above and below the default has been used to determine the minimum and maximum of all the parameters. However, this percentage could be specific to each parameter to accommodate for the real variation occurring naturally in biological systems.
- In our work, the input data that we give to our NN are fixed. The matrix type n_{types} is equal to 4 for the matrix type and the maximum number of draws n_{draws} is equal to 8. A possible improvement is to increase n_{draws} by reducing the interval between the different timesteps. An interval of 24 hours compared to the actual 100 hours could lead to better performance and an ART process more inline with the interval between RT fractions. New matrix type could also be drawn from the simulation to give more information to our model. It is important to note that this increase in input data is resource-intensive for the model.
- The methodology used to predict the parameters could also be discussed. The utilisation of a neural network was not much debated due to the performance of these models in prediction tasks. However, other methods such as Support Vector Machine (SVM), tree-based models, and K-Nearest Neighbour (KNN) could be implemented and compared with the performance of our NN. Improvements might also be possible with regard to the architecture and the training of the NN. The current method is to retrain the model entirely each time the number of draws increase. An enhancement to this process is to reuse the weight of the previous CNN pipelines, either by freezing them and train only the rest of the NN or to re-train the whole model but with these weights as initialisation.
- The performance of our model could benefit by further refining the hyperparameter tuning process. Our models possess currently 8 hyperparameters, and while they cover the main part of our architecture and our training method, they are not exhaustive. Others hyperparameters specific to the CNN layers (kernel size) or the activation function could be tweaked to improve the performance. Other algorithms for the hyperparameters search have also been considered like a local search or grid search. Actually, a local search based approach has been tested but not kept due to time limitation.

Chapter 8

Conclusion

The main purpose of this master thesis is to construct a parameter identification pipeline able to predict the initial parameters of a tumour growth cellular model. Through our work, we constructed three datasets containing the outputs matrices of the simulation: one dataset without RT, one with the baseline treatment, and one with the optimised treatment from Martin et al. [2]. These datasets have been explored with dimensionality reduction and mutual information to obtain valuable insights about both the behaviour of the tumour model and the future prediction performance.

Our prediction model is a hybrid NN architecture based on the utilisation of both CNN and LSTM. Its performance is very heterogeneous between the five parameters we are trying to predict.

- **The duration of the cell cycle** is the easiest parameter to reconstruct and achieves an error of only 5%, which is equivalent to 1.5 hours. These results were highly anticipated due to the direct impact this parameter has on the simulation.
- The performances of **the average consumptions of both oxygen and glucose in cancer cells** are equivalent. They achieve a prediction error of around 17%. On the dataset without treatment, the error decreases significantly over time.
- **The average glucose absorptions of healthy cells** has very scattered results with respect to time and dataset. The error varies between 10% and 25%.
- Finally, the model produces surprising results when predicting **the average oxygen consumption of healthy cells**. The mean error in prediction is of $8-9\%$. These results were not expected but are promising for our research.

On top of that, in an effort to improve our interpretation of the performance, we studied the inherent similarity between matrices. It enables us to free ourselves from the stochastic nature of the simulation. In practice, we observed how the prediction error impacts the difference between matrices. This analysis has revealed to us that even for low prediction error in parameter, and inherent error in certain matrices may still be present, leading to the conclusion that our model might suffer from performance limitations.

All things considered, the research conducted in this master thesis demonstrates that effective performance could be achieved in this task but not on all the parameters. Our work shows evidence that potential further works exist in adapting a cellular simulation to a real tumour anatomy.

Chapter 9

Annexe

9.1 Full dimensional reduction with PCA algorithm

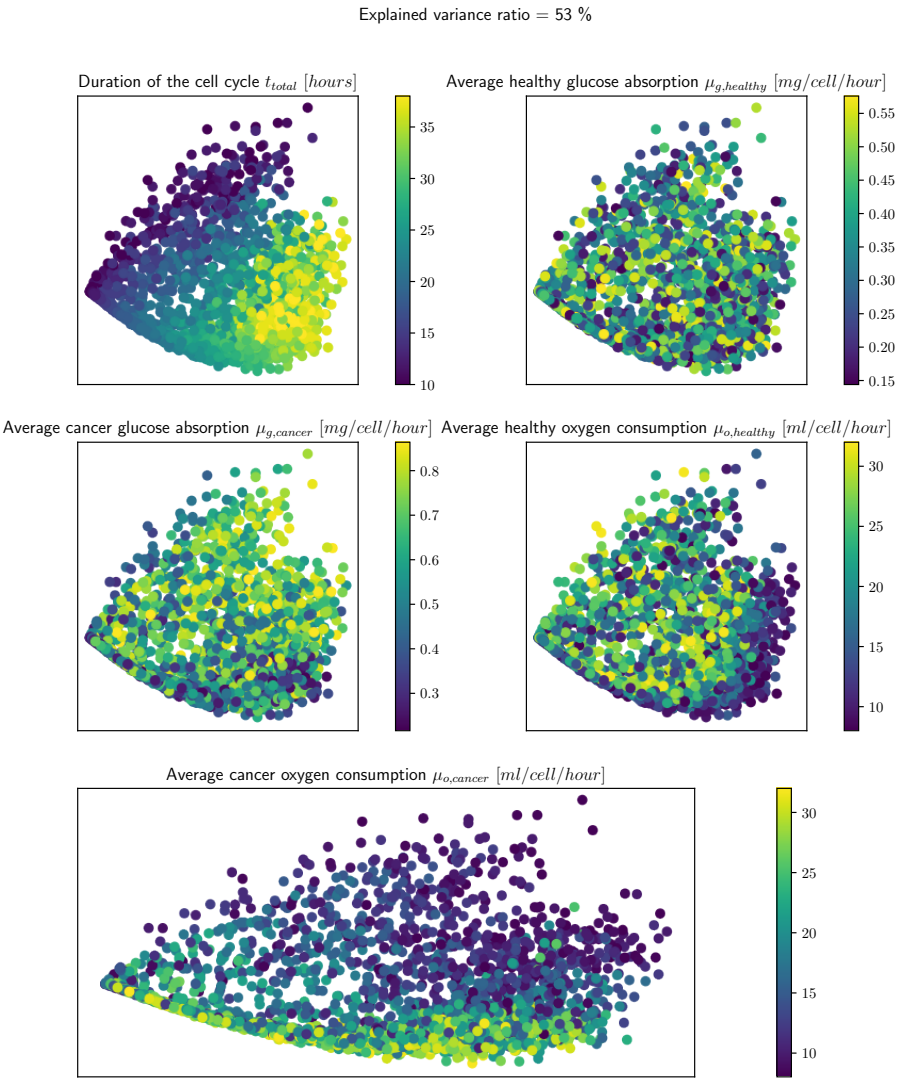

Figure 9.1: Dimensional reduction of the dataset without treatment using PCA. Parameters to predict are shown with the color gradient

9.1.1 Duration of the cell cycle (t_{total})

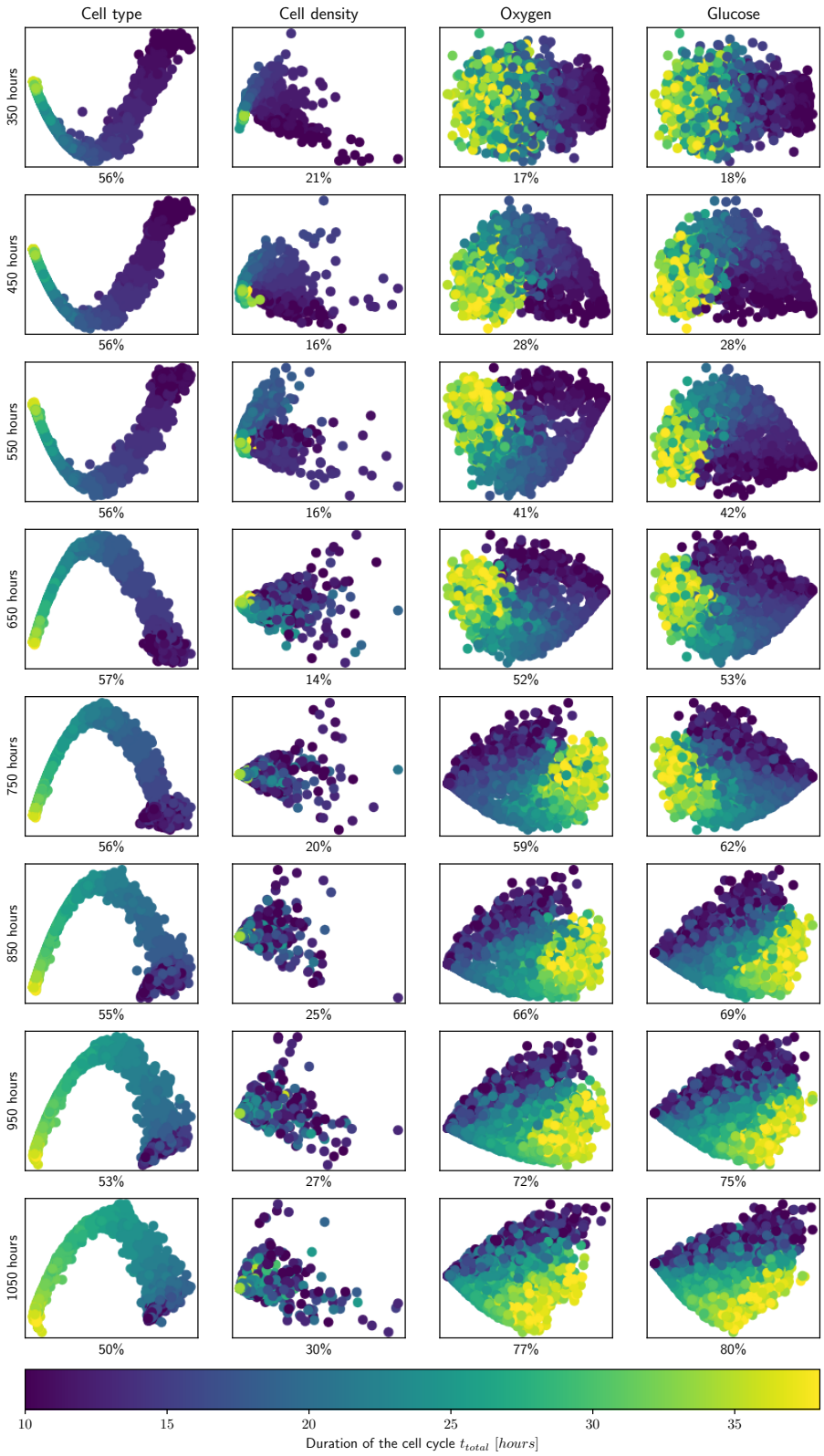

Figure 9.2: Dimensional reduction using PCA of the different images in the dataset without treatment highlighting the heterogeneity of the duration of the cell cycle. The explained variance ratio of the PCA is displayed below each graph

9.1.2 Average glucose absorptions of cancers cells ($\mu_{g,cancer}$)

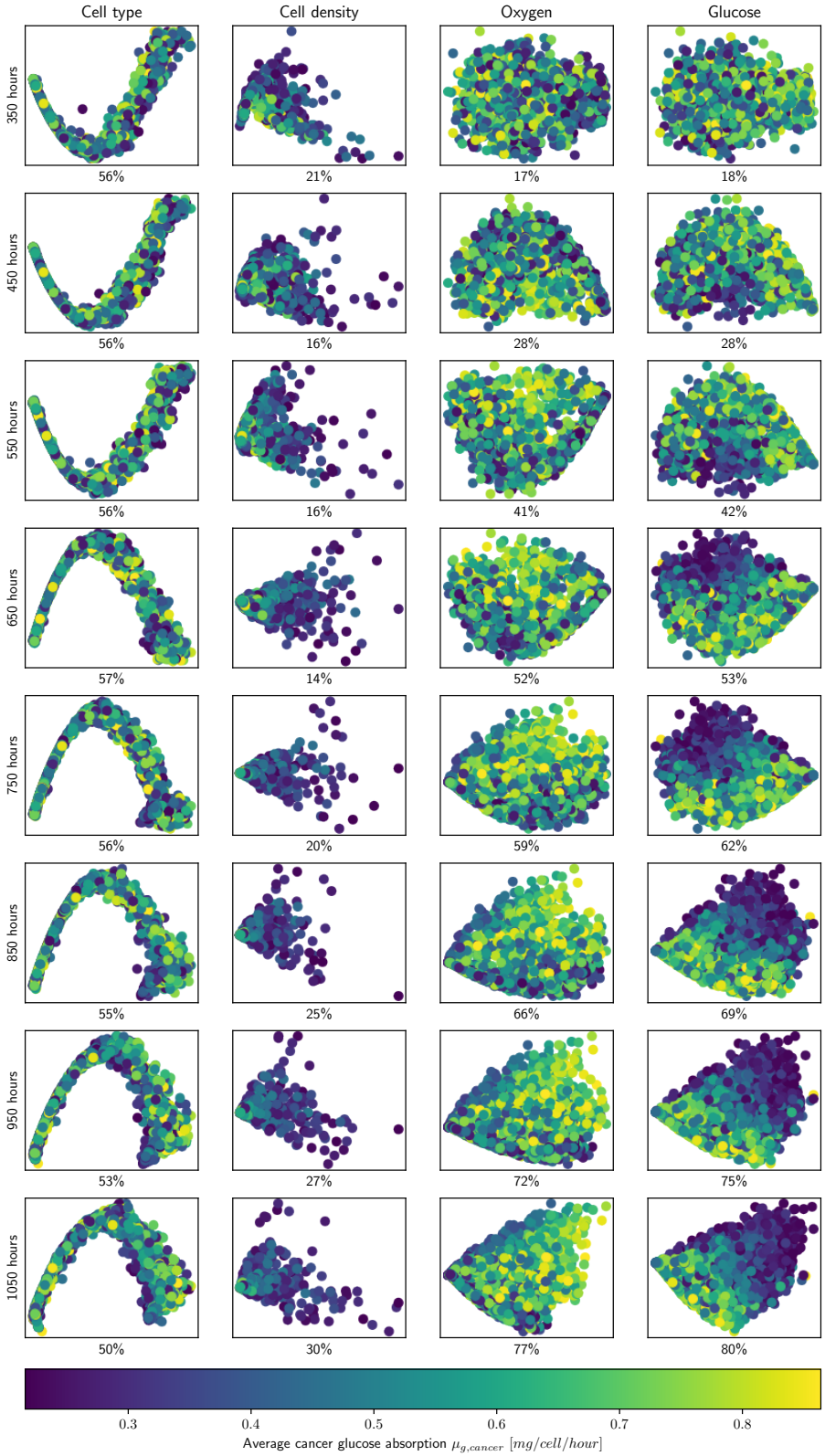

Figure 9.3: Dimensional reduction using PCA of the different images in the dataset without treatment highlighting the heterogeneity of the average glucose absorption of cancer cells. The explained variance ratio of the PCA is displayed below each graph

9.1.3 Average oxygen consumptions of cancers cells ($\mu_{o,cancer}$)

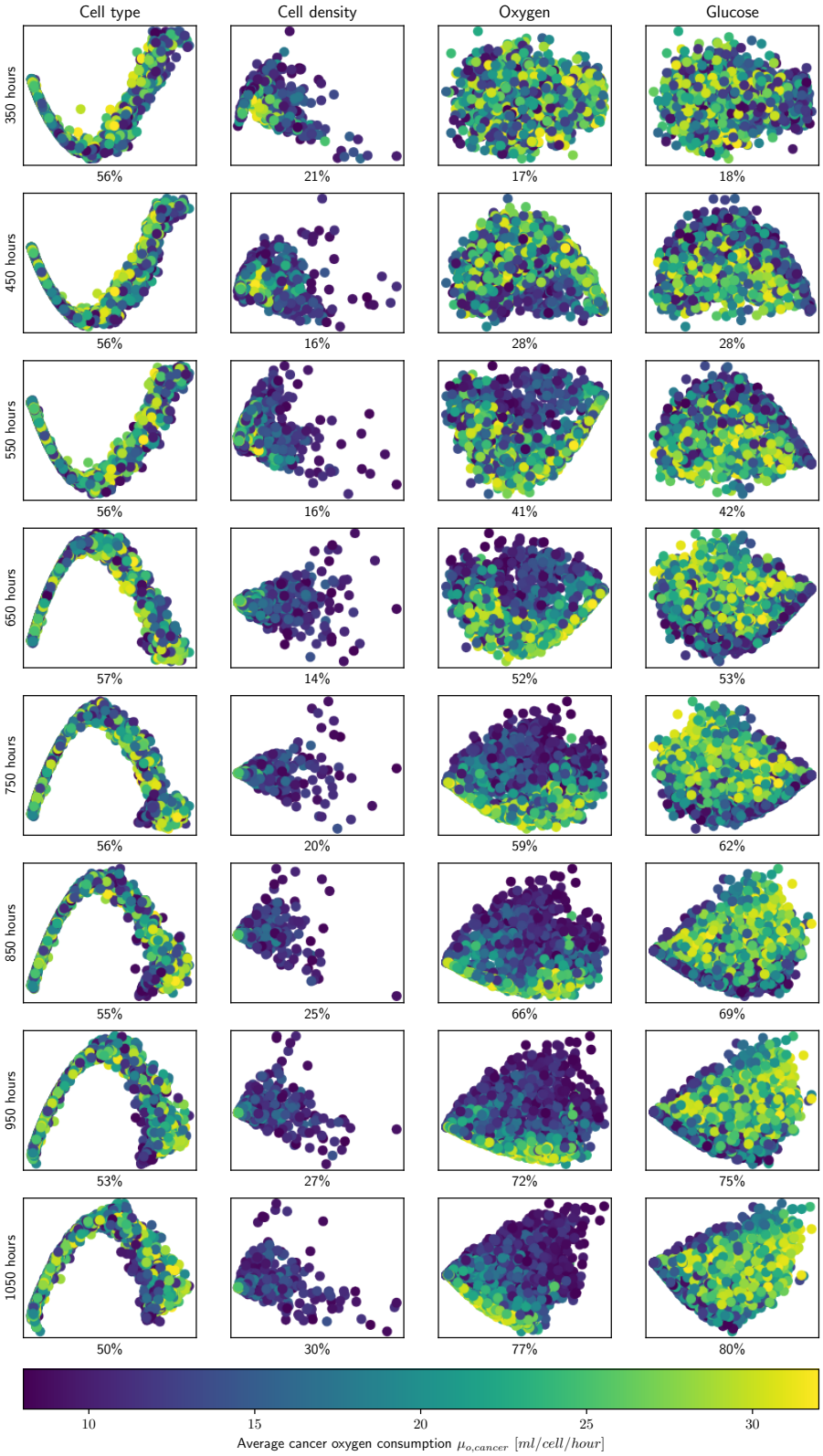

Figure 9.4: Dimensional reduction using PCA of the different images in the dataset without treatment highlighting the heterogeneity of the average oxygen consumption of cancer cells. The explained variance ratio of the PCA is displayed below each graph

9.1.4 Average glucose absorptions of healthy cells ($\mu_{g,healthy}$)

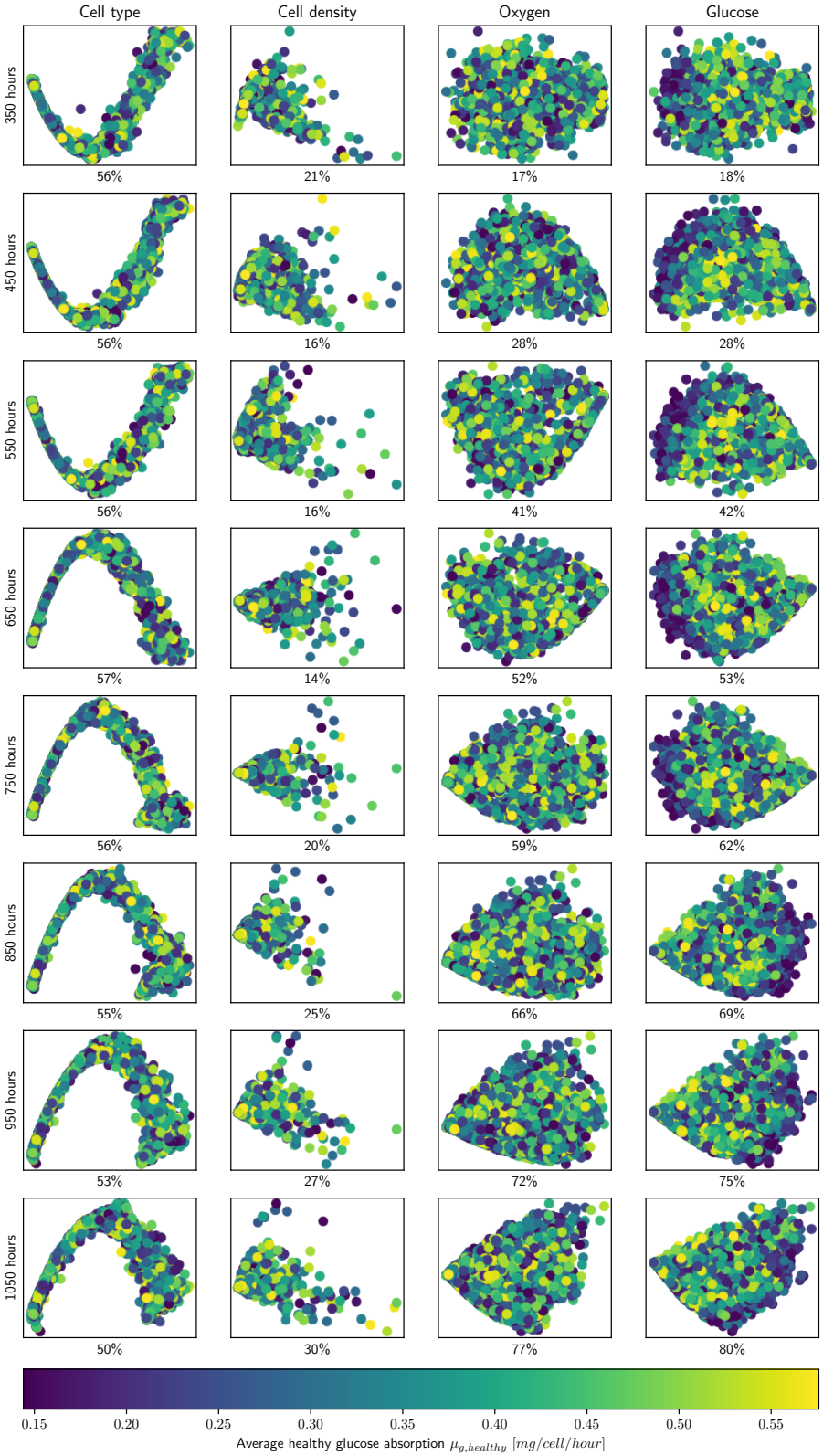

Figure 9.5: Dimensional reduction using PCA of the different images in the dataset without treatment highlighting the heterogeneity of the average glucose absorption of healthy cells. The explained variance ratio of the PCA is displayed below each graph

9.1.5 Average oxygen consumptions of healthy cells ($\mu_{o,healthy}$)

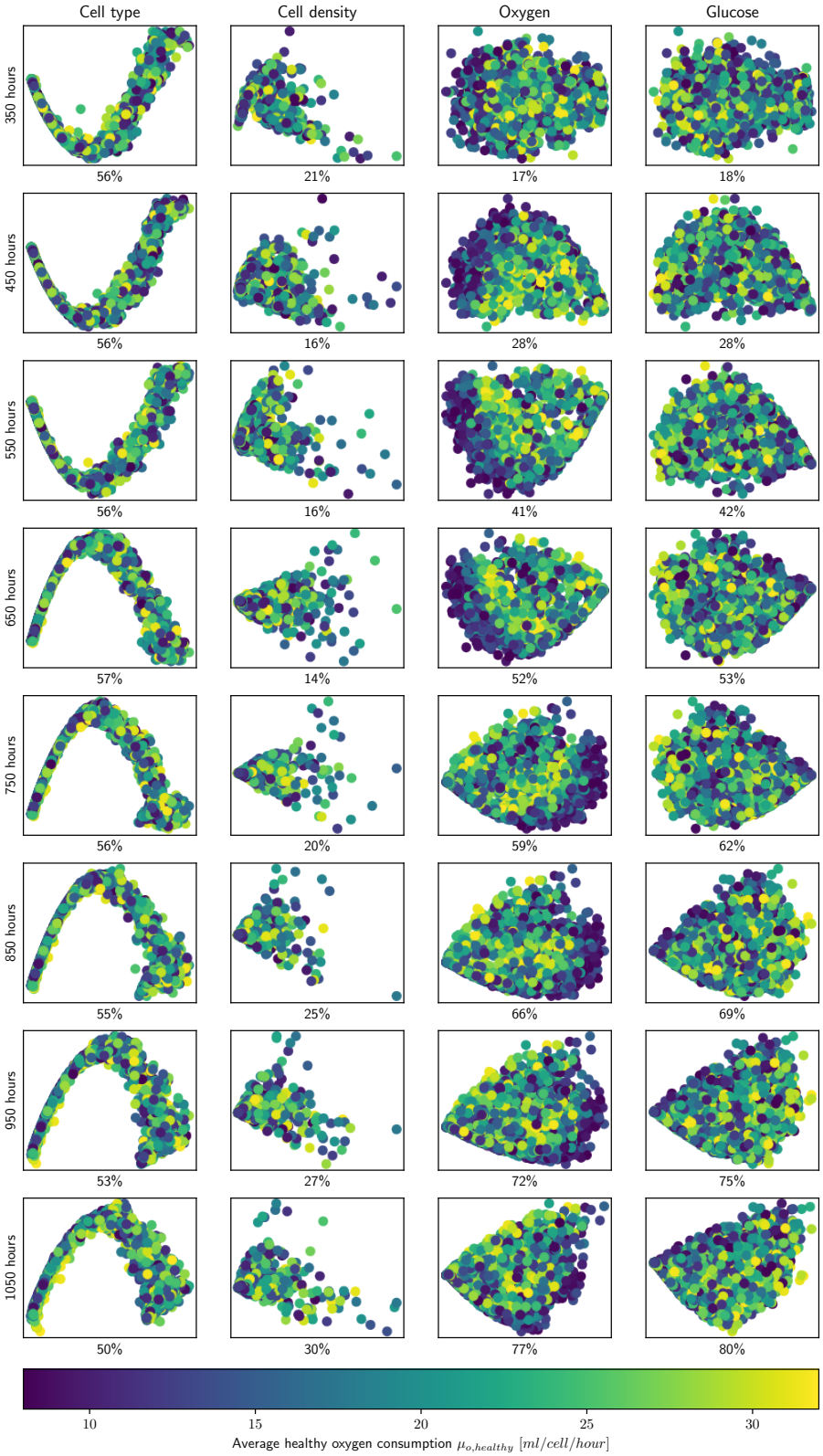

Figure 9.6: Dimensional reduction using PCA of the different images in the dataset without treatment highlighting the heterogeneity of the average oxygen consumption of healthy cells. The explained variance ratio of the PCA is displayed below each graph

9.2 Full dimensional reduction with Isomap algorithm

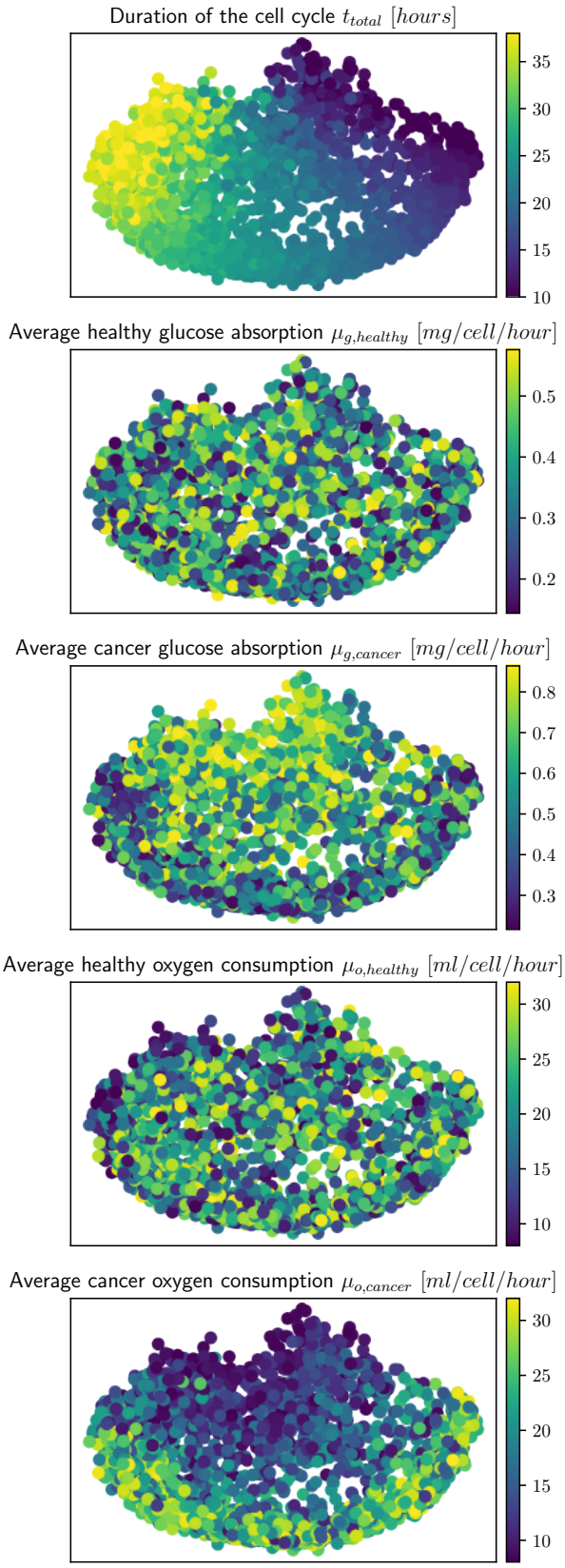

Figure 9.7: Dimensional reduction of the dataset without treatment using isomap. Parameters to predict are shown with the color gradient

9.2.1 Duration of the cell cycle (t_{total})

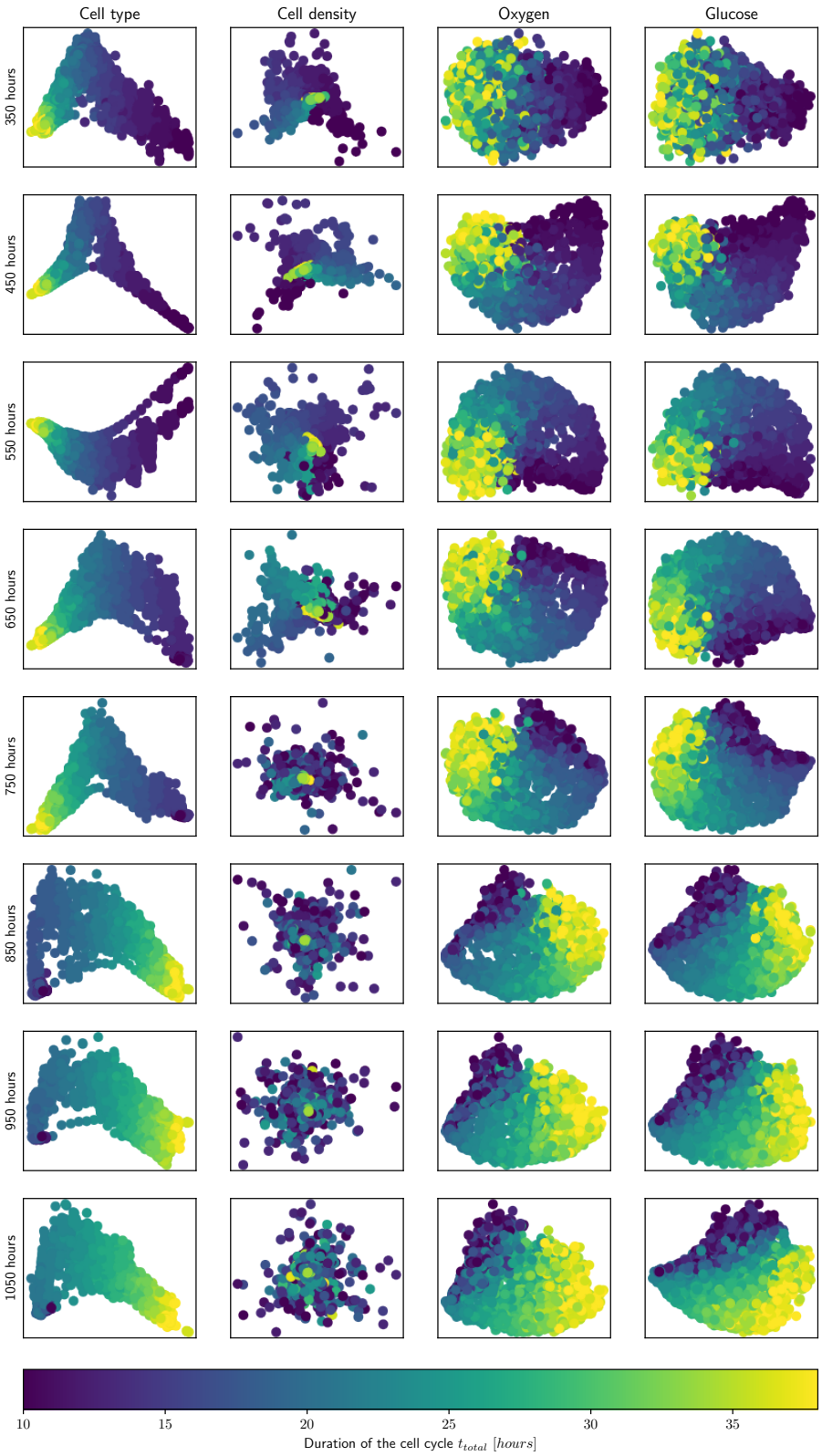

Figure 9.8: Dimensional reduction using isomap of the different images in the dataset without treatment highlighting the heterogeneity of the duration of the cell cycle.

9.2.2 Average glucose absorptions of cancers cells ($\mu_{g,cancer}$)

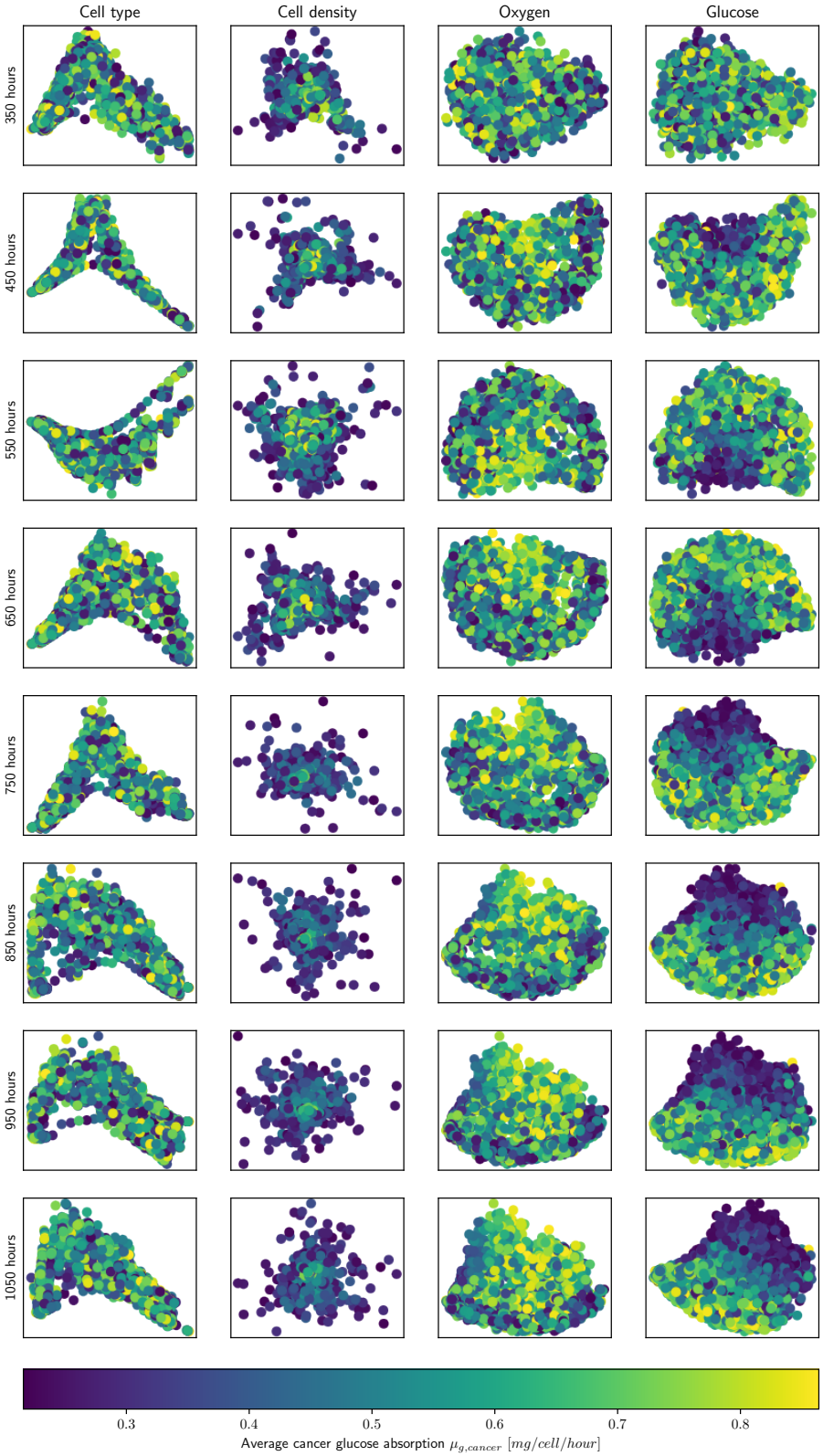

Figure 9.9: Dimensional reduction using isomap of the different images in the dataset without treatment highlighting the heterogeneity of the average glucose absorption of cancer cells.

9.2.3 Average oxygen consumptions of cancers cells ($\mu_{o,cancer}$)

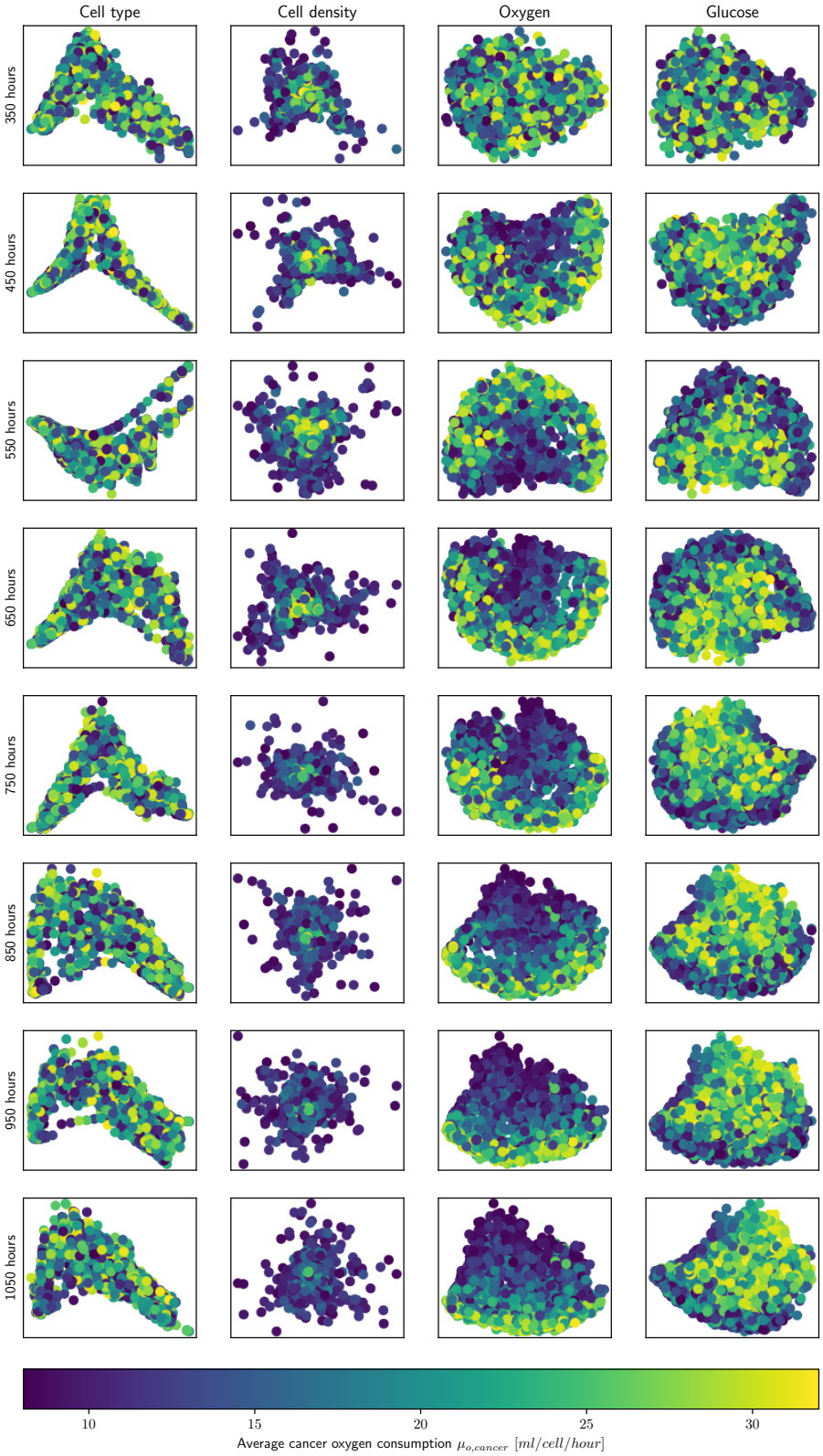

Figure 9.10: Dimensional reduction using isomap of the different images in the dataset without treatment highlighting the heterogeneity of the average oxygen consumption of cancer cells.

9.2.4 Average glucose absorptions of healthy cells ($\mu_{g,healthy}$)

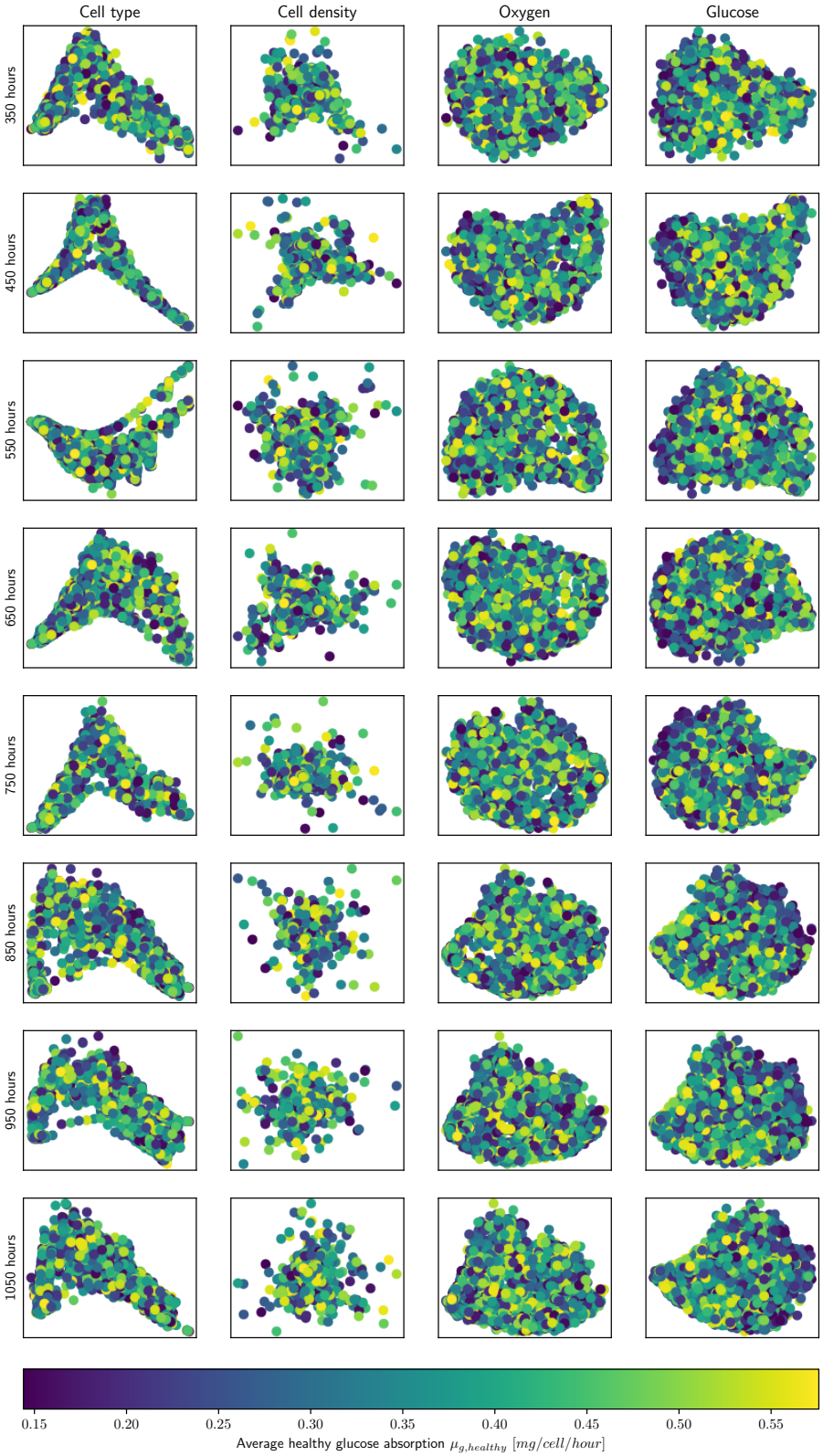

Figure 9.11: Dimensional reduction using isomap of the different images in the dataset without treatment highlighting the heterogeneity of the average glucose absorption of healthy cells.

9.2.5 Average oxygen consumptions of healthy cells ($\mu_{o,healthy}$)

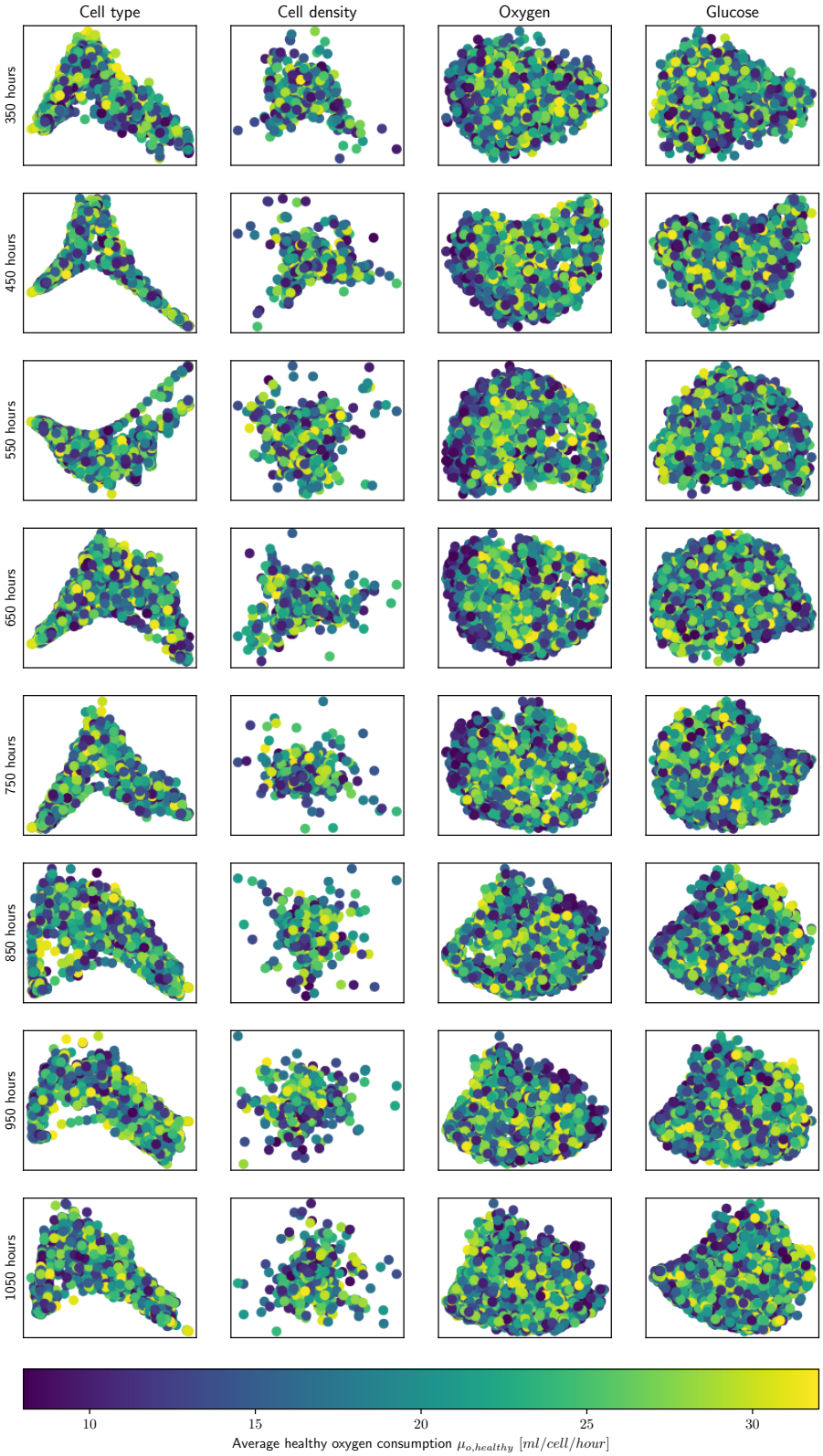

Figure 9.12: Dimensional reduction using isomap of the different images in the dataset without treatment highlighting the heterogeneity of the average oxygen consumption of healthy cells.

9.3 Full mutual information between matrices and target

9.3.1 Duration of the cell cycle (t_{total})

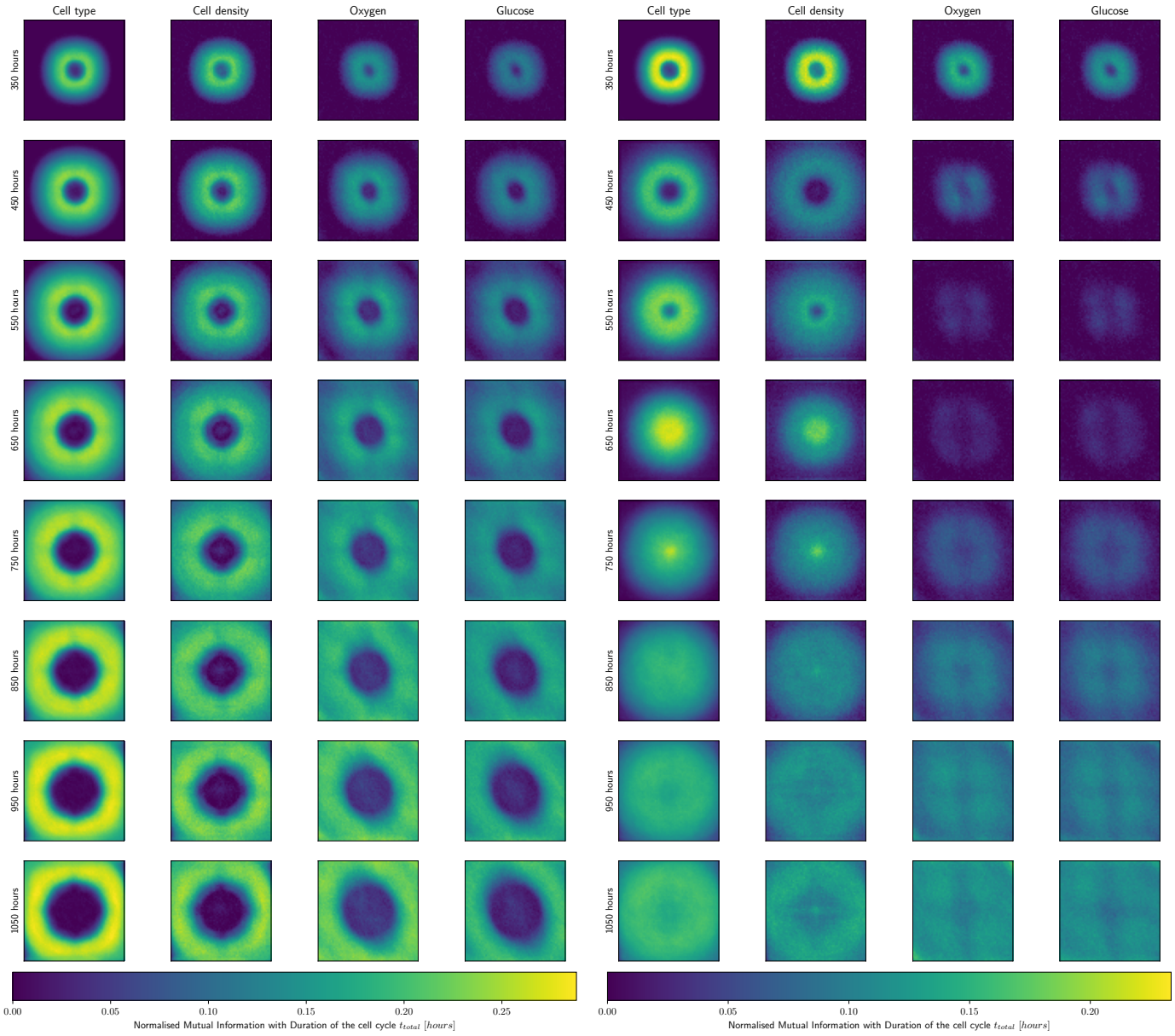

Figure 9.13: Mutual information between matrices and the duration of the cell cycle on the dataset without treatment (left) and with treatment (right).

9.3.2 Average glucose absorptions of cancers cells ($\mu_{g,cancer}$)

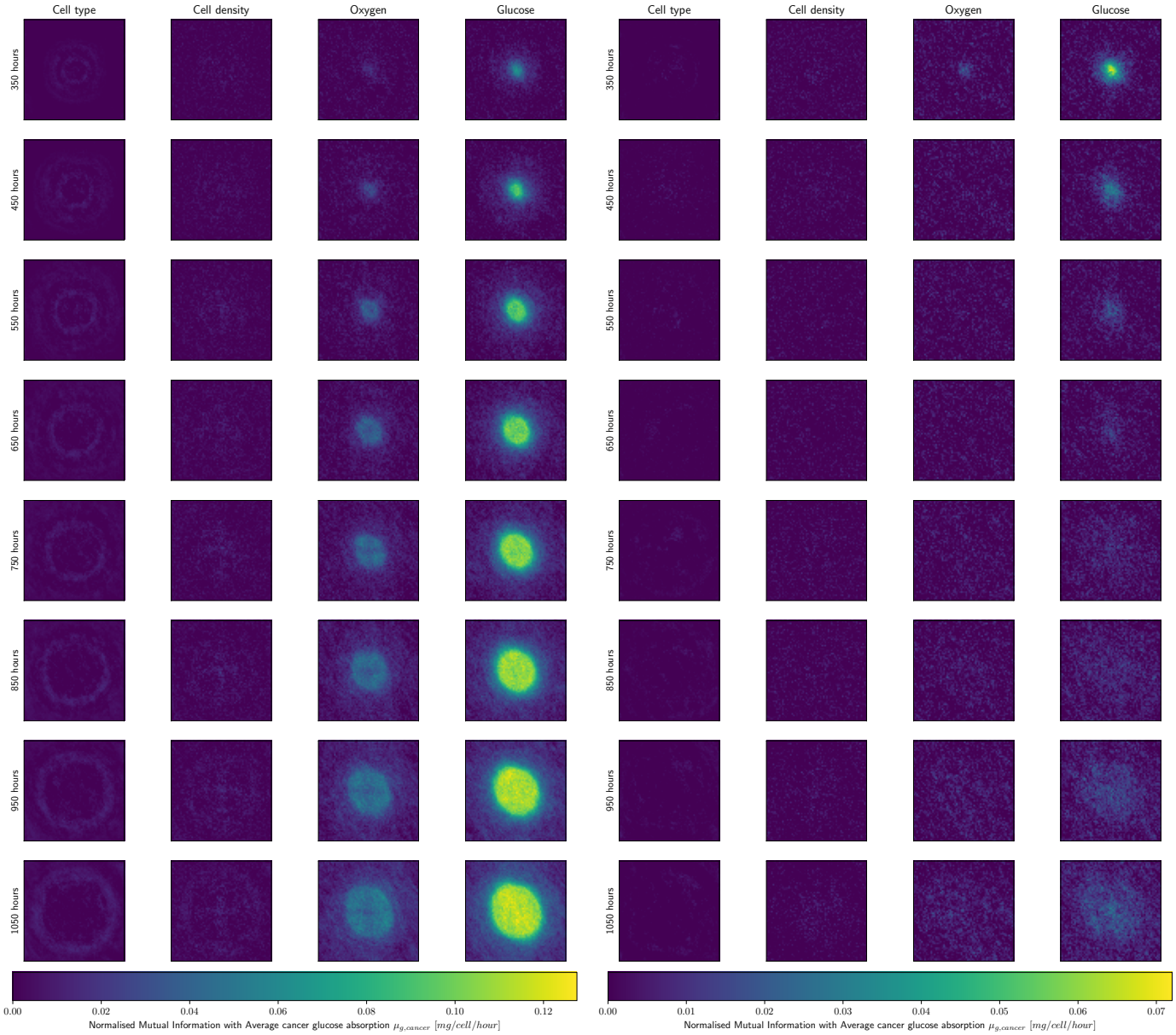

Figure 9.14: Mutual information between matrices and the average glucose absorption of cancers cells on the dataset without treatment (left) and with treatment (right).

9.3.3 Average oxygen consumptions of cancers cells ($\mu_{o,cancer}$)

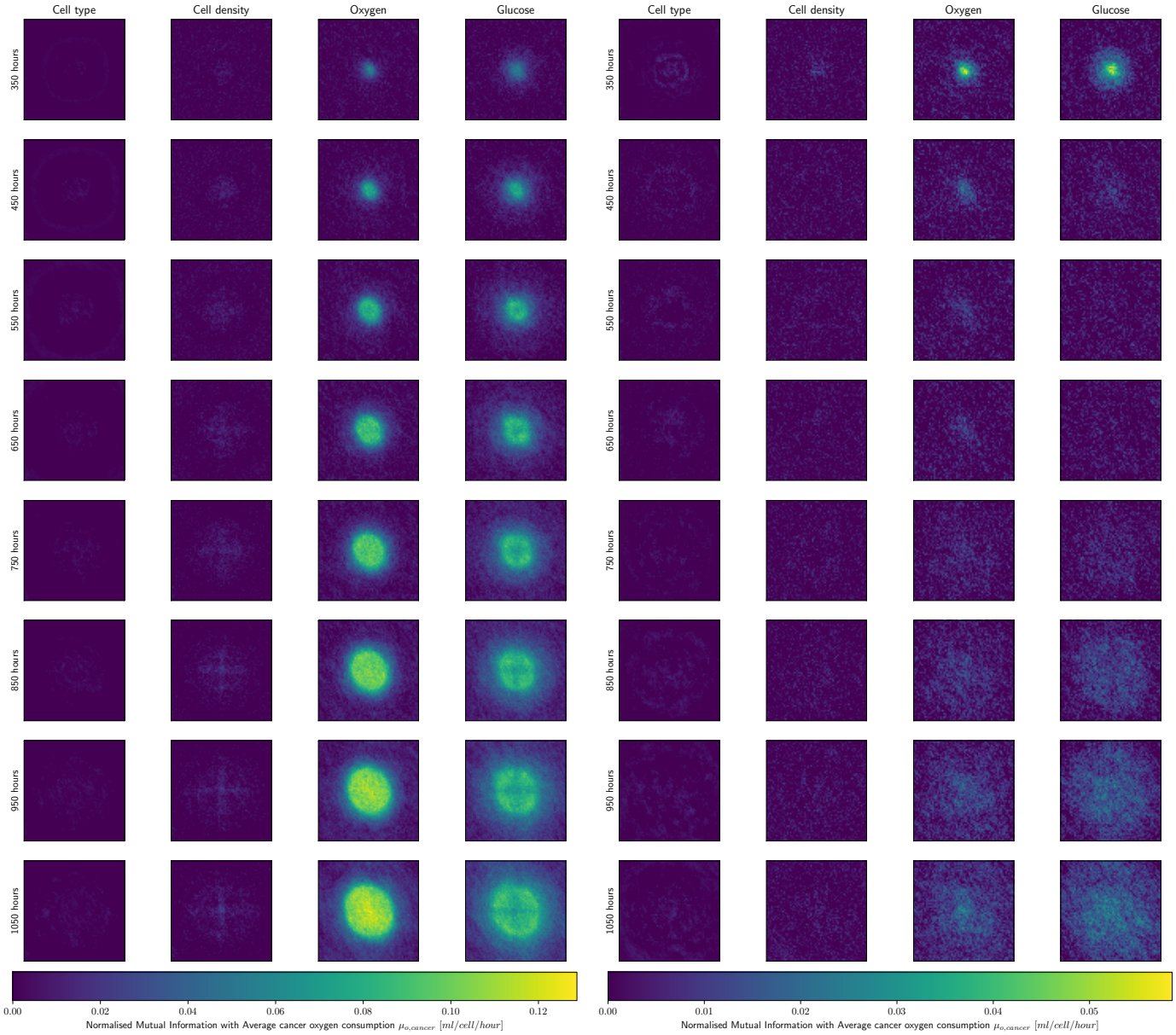

Figure 9.15: Mutual information between matrices and the average oxygen consumption of cancers cells on the dataset without treatment (left) and with treatment (right).

9.3.4 Average glucose absorptions of healthy cells ($\mu_{g,healthy}$)

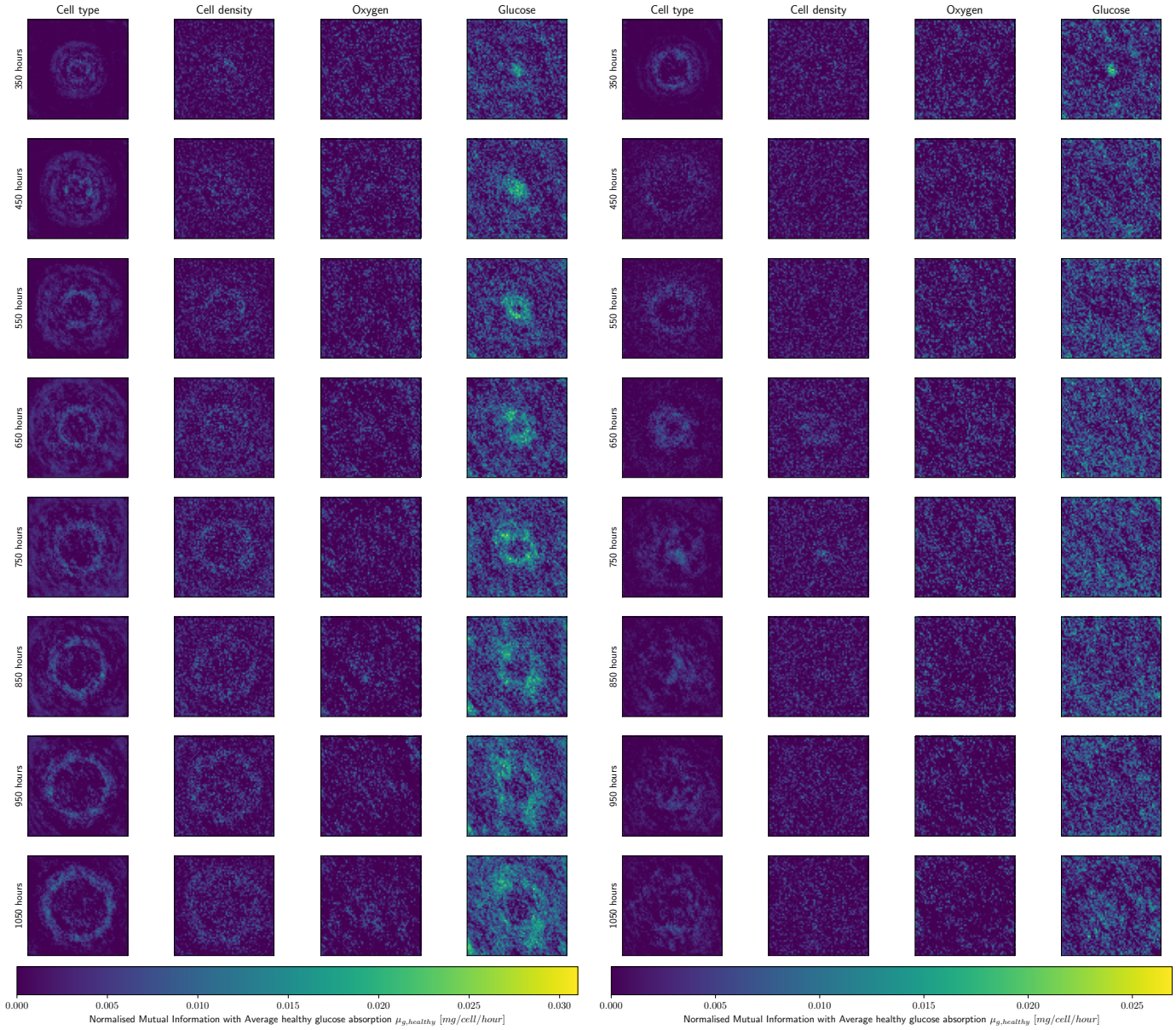

Figure 9.16: Mutual information between matrices and the average glucose absorption of healthy cells on the dataset without treatment (left) and with treatment (right).

9.3.5 Average oxygen consumptions of healthy cells ($\mu_{o,healthy}$)

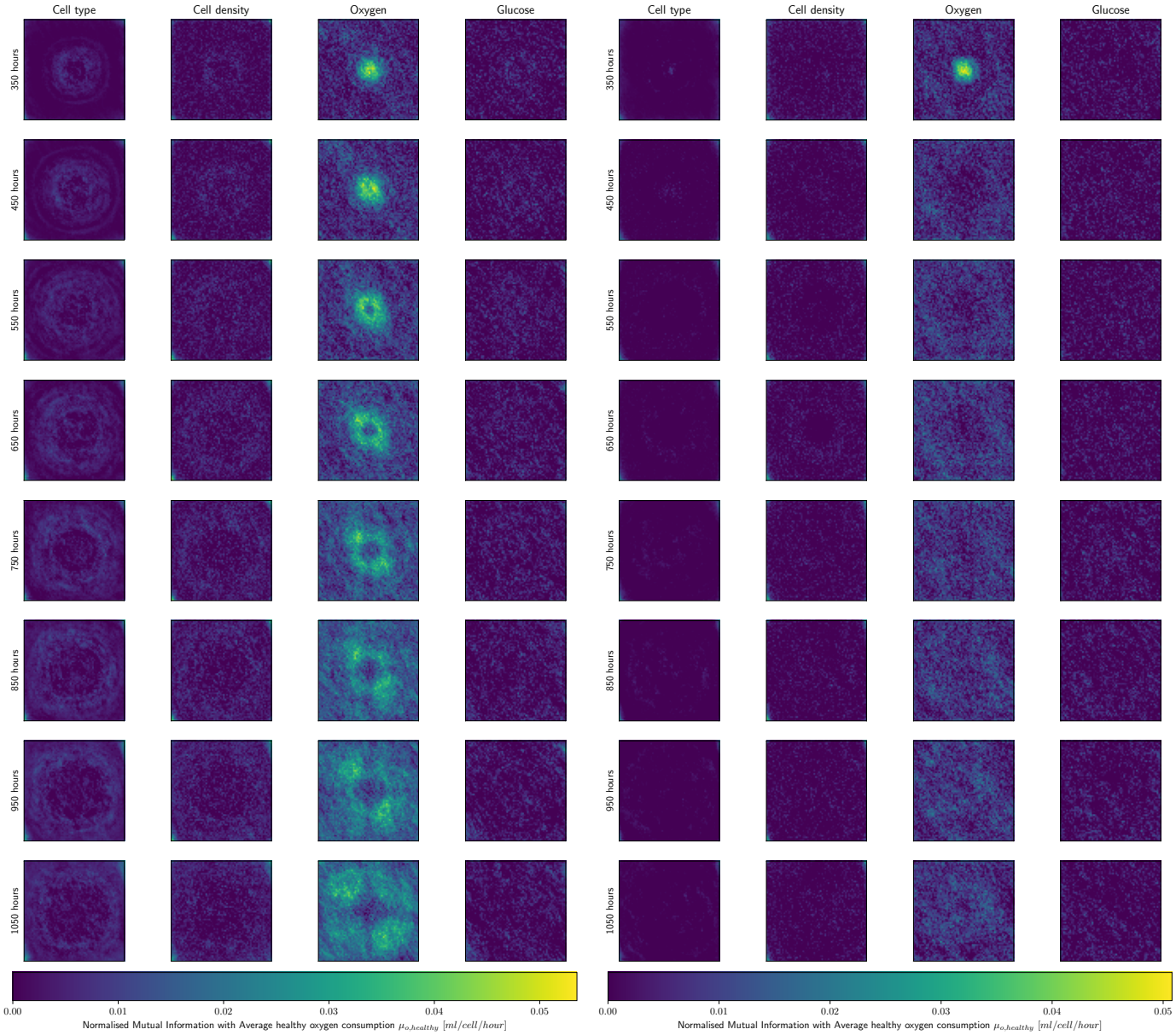

Figure 9.17: Mutual information between matrices and the average oxygen consumption of healthy cells on the dataset without treatment (left) and with treatment (right).

9.4 Optimal hyperparameters

Draws	$n_{in,LSTM}$	$n_{out,LSTM}$	n_{LSTM}	n_{conv}	Learning Rate	Batch Size	L2 Regularization	$n_{feed\ forward}$
1	500	200	0	2	0.0001	4	0.001	500
2	1000	100	0	4	0.0001	8	0.005	500
3	500	1000	0	4	5e-05	4	0.005	1000
4	500	100	1	3	1e-05	16	0.001	200
5	200	100	0	4	5e-05	1	0.01	200
6	500	500	1	4	0.0001	16	0.001	500
7	1000	500	2	4	0.0001	32	0.005	0
8	100	500	1	4	5e-05	8	0.005	0

Table 9.1: Optimal hyperparameters per number of draws on the dataset without treatment.

Draws	$n_{in,LSTM}$	$n_{out,LSTM}$	n_{LSTM}	n_{conv}	Learning Rate	Batch Size	L2 Regularization	$n_{feed\ forward}$
1	100	200	1	3	5e-05	4	0.01	0
2	500	1000	1	4	0.0001	8	0.001	0
3	500	100	1	2	5e-05	4	0.01	200
4	200	500	0	4	5e-05	4	0.001	100
5	200	200	2	3	0.0001	32	0.001	200
6	200	500	2	4	0.0001	32	0.001	500
7	1000	1000	3	4	0.0001	64	0.001	0
8	1000	1000	1	2	0.0001	16	0.01	1000

Table 9.2: Optimal hyperparameters per number of draws on the dataset with baseline treatment.

Draws	$n_{in,LSTM}$	$n_{out,LSTM}$	n_{LSTM}	n_{conv}	Learning Rate	Batch Size	L2 Regularization	$n_{feed\ forward}$
1	100	500	0	3	5e-05	32	0.01	100
2	1000	1000	1	4	0.0001	16	0.001	500
3	200	100	1	4	5e-05	8	0.001	200
4	500	500	2	3	5e-06	64	0.001	0
5	200	200	2	4	5e-05	8	0.001	100
6	500	200	0	1	5e-05	2	0.001	200
7	500	200	1	2	5e-05	2	0.001	200
8	500	1000	2	4	0.0005	32	0.001	0

Table 9.3: Optimal hyperparameters per number of draws on the dataset with RL treatment.

9.5 Hyperparameters influence

9.5.1 Learning rate

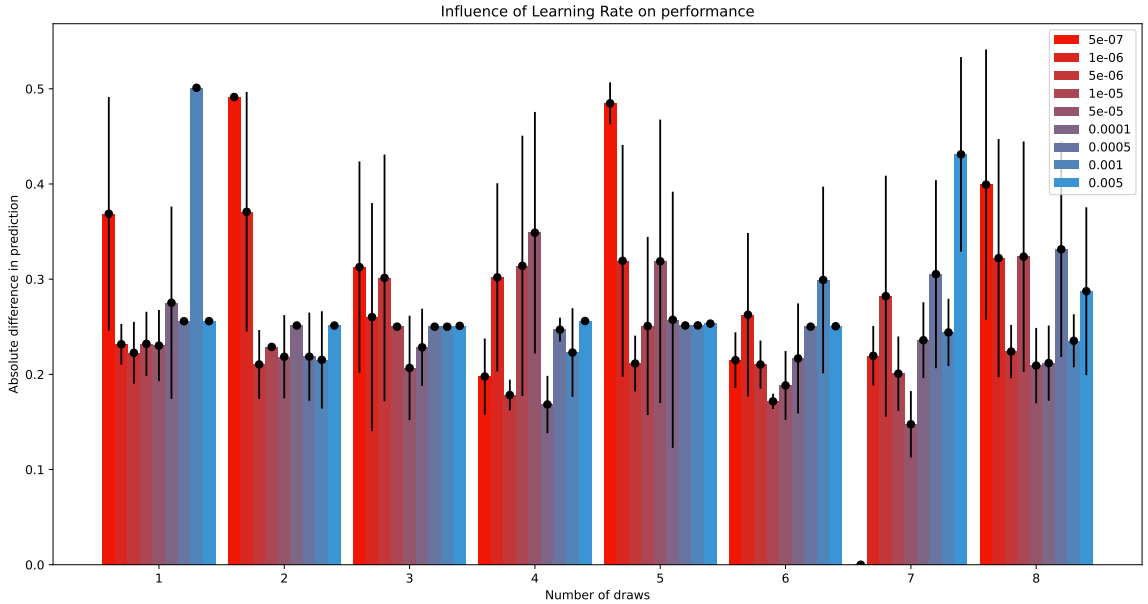

Figure 9.18: Mean and standard deviation of the performance for different learning rate

9.5.2 Batch size

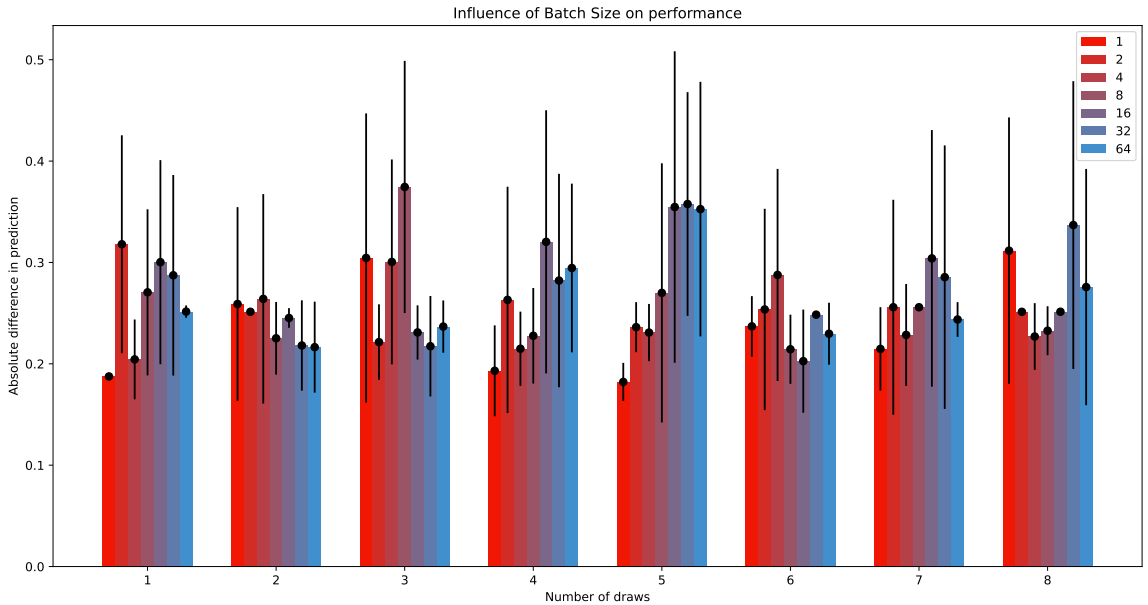

Figure 9.19: Mean and standard deviation of the performance for different batch size

9.5.3 L2 regularization

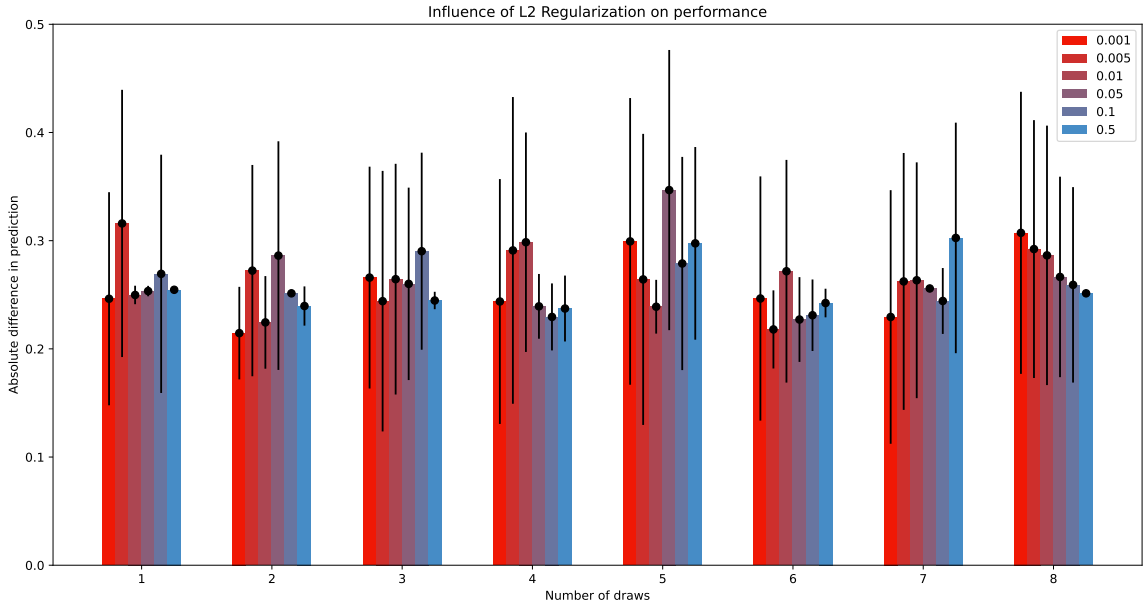

Figure 9.20: Mean and standard deviation of the performance for different value of the L2 regularization

9.5.4 Number of alternating layers of convolution and max-pooling n_{conv}

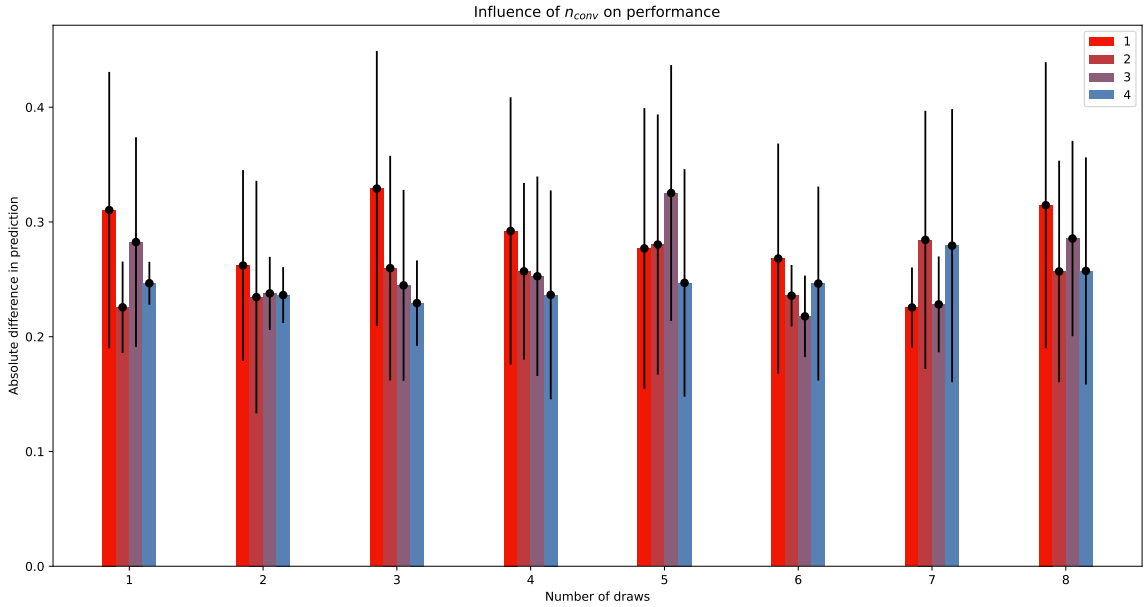

Figure 9.21: Mean and standard deviation of the performance for different number of convolution layers

9.5.5 Number of LSTM layers n_{LSTM}

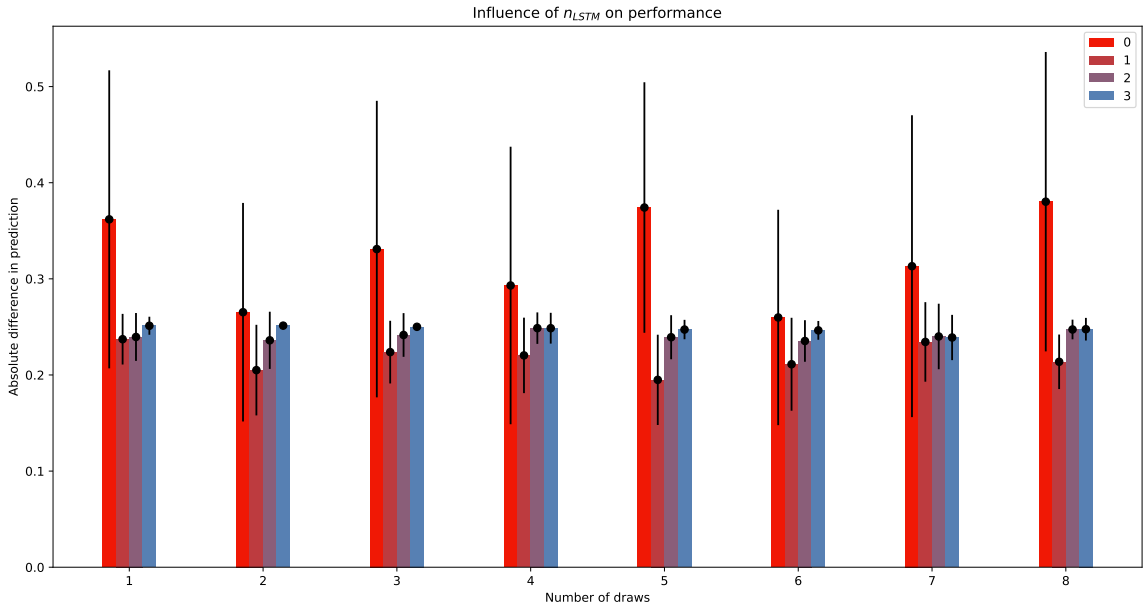

Figure 9.22: Mean and standard deviation of the performance for different number of LSTM layers

9.5.6 Size of the encoded vector that goes into each LSTM blocks $n_{in,LSTM}$

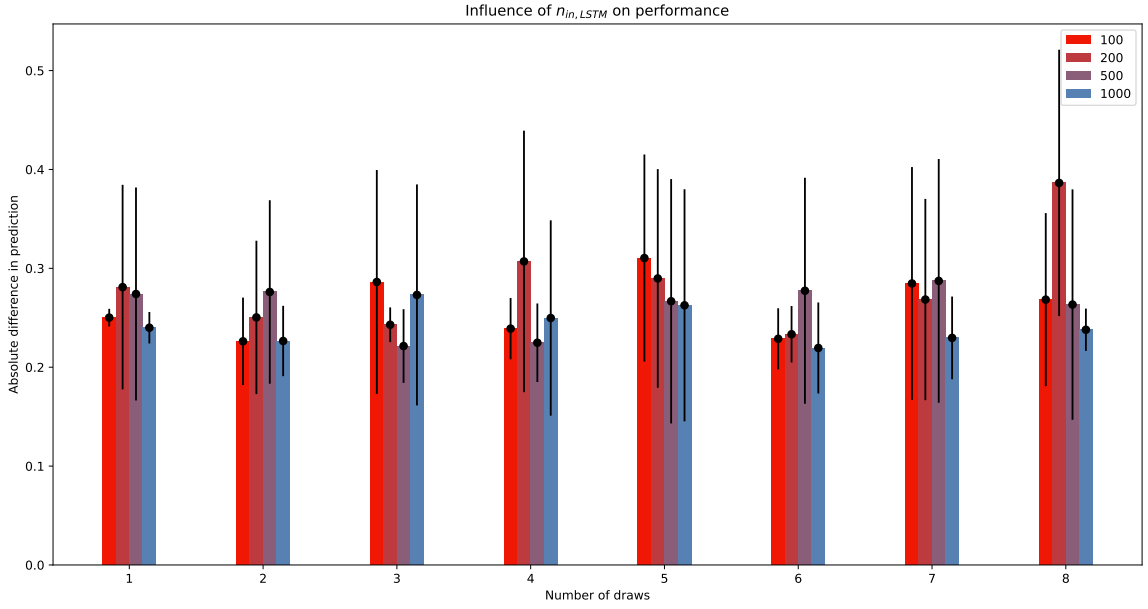

Figure 9.23: Mean and standard deviation of the performance for different size of the encoded vector before the LSTM

9.5.7 Size of the output of each LSTM blocks $n_{out,LSTM}$

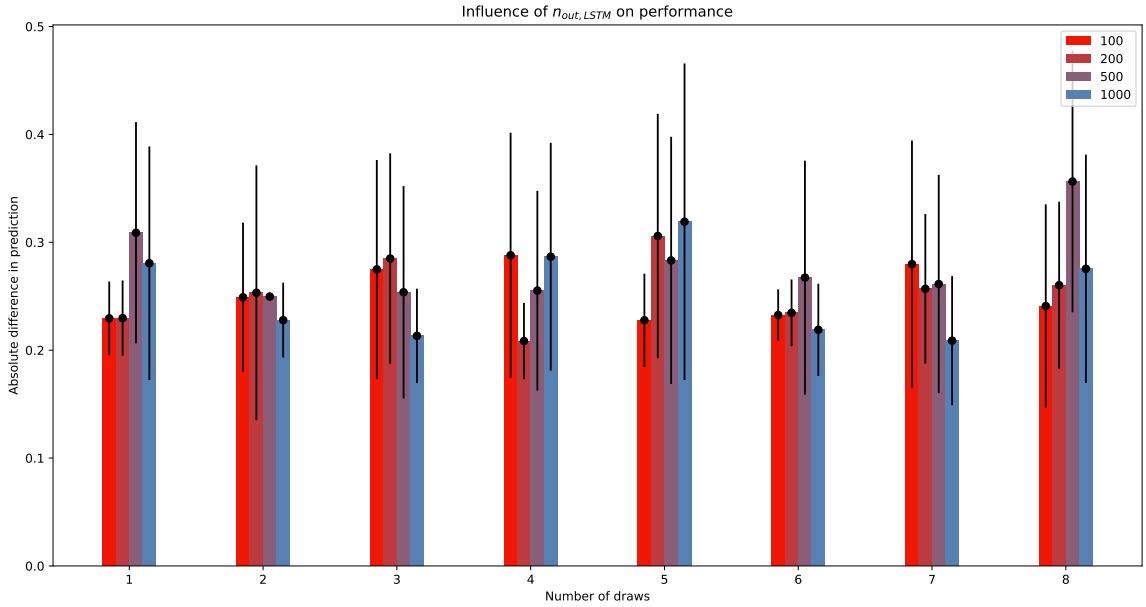

Figure 9.24: Mean and standard deviation of the performance for different size of the output vector after the LSTM

9.5.8 Number of neurons in the final hidden layer $n_{feed\ forward}$

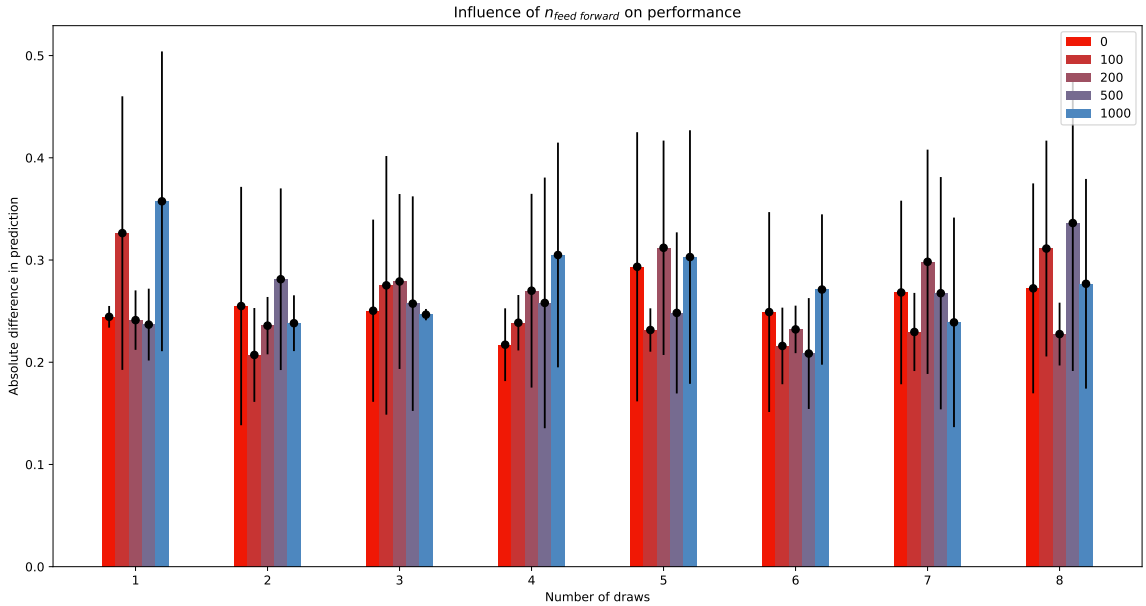

Figure 9.25: Mean and standard deviation of the performance for different number of neurons in the final hidden layer

Bibliography

[1] Grégoire Moreau et al. "Reinforcement Learning for Radiotherapy Dose Fractioning Automation". en. In: *Biomedicines* 9.2 (Feb. 2021). Number: 2 Publisher: Multidisciplinary Digital Publishing Institute, p. 214. ISSN: 2227-9059. DOI: 10.3390/biomedicines9020214. URL: https://www.mdpi.com/2227-9059/9/2/214 (visited on 05/17/2024).

[2] Florian Martin. "Robust Reinforcement Learning optimization of radiotherapy dose fractioning". PhD thesis. 2023.

[3] Ammar Jalalimanesh et al. "Simulation-based optimization of radiotherapy: Agent-based modeling and reinforcement learning". In: *Mathematics and Computers in Simulation*. Biomath 2014 and Biomath 2015 133 (Mar. 2017), pp. 235–248. ISSN: 0378-4754. DOI: 10.1016/j.matcom.2016.05.008. URL: https://www.sciencedirect.com/science/article/pii/S0378475416300878 (visited on 05/17/2024).

[4] Nicole O'Neil. "An Agent Based Model of Tumor Growth and Response to Radiotherapy". In: *Theses and Dissertations* (May 2012). DOI: https://doi.org/10.25772/ZWX8-SN12. URL: https://scholarscompass.vcu.edu/etd/2831.

[5] B. D. Editors. *Cell Cycle - Definition, Phases, Examples, Regulation*. en-US. May 2017. URL: https://biologydictionary.net/cell-cycle/ (visited on 04/08/2024).

[6] *What Is Cancer? - NCI*. en. cgvArticle. Archive Location: nciglobal,ncienterprise. Sept. 2007. URL: https://www.cancer.gov/about-cancer/understanding/what-is-cancer (visited on 04/08/2024).

[7] *External Beam Radiation Therapy for Cancer - NCI*. en. cgvArticle. Archive Location: nciglobal,ncienterprise. May 2018. URL: https://www.cancer.gov/about-cancer/treatment/types/radiation-therapy/external-beam (visited on 04/25/2024).

[8] H. B. Mitchell. "Image Similarity Measures". en. In: *Image Fusion: Theories, Techniques and Applications*. Ed. by H. B. Mitchell. Berlin, Heidelberg: Springer, 2010, pp. 167–185. ISBN: 978-3-642-11216-4. DOI: 10.1007/978-3-642-11216-4_14. URL: https://doi.org/10.1007/978-3-642-11216-4_14 (visited on 07/26/2024).

[9] Michel Marie Deza and Elena Deza. "Distances and Similarities in Data Analysis". en. In: *Encyclopedia of Distances*. Ed. by Michel Marie Deza and Elena Deza. Berlin, Heidelberg: Springer, 2013, pp. 291–305. ISBN: 978-3-642-30958-8. DOI: 10.1007/978-3-642-30958-8_17. URL: https://doi.org/10.1007/978-3-642-30958-8_17 (visited on 07/30/2024).

[10] Vasista Reddy. *Exploring Image Similarity Approaches in Python*. en. Oct. 2023. URL: https://medium.com/scrapehero/exploring-image-similarity-approaches-in-python-b8ca0a3ed5a3 (visited on 07/30/2024).

[11] Z. Wang et al. "Image Quality Assessment: From Error Visibility to Structural Similarity". en. In: *IEEE Transactions on Image Processing* 13.4 (Apr. 2004), pp. 600–612. ISSN: 1057-7149. DOI: 10.1109/TIP.2003.819861. URL: http://ieeexplore.ieee.org/document/1284395/ (visited on 07/30/2024).

[12] Philipp M. Altrock, Lin L. Liu, and Franziska Michor. "The mathematics of cancer: integrating quantitative models". en. In: *Nature Reviews Cancer* 15.12 (Dec. 2015). Publisher: Nature Publishing Group, pp. 730–745. ISSN: 1474-1768. DOI: 10.1038/nrc4029. URL: https://www.nature.com/articles/nrc4029 (visited on 05/05/2024).

[13] Lucas B. Edelman, James A. Eddy, and Nathan D. Price. "In silico models of cancer". en. In: *WIREs Systems Biology and Medicine* 2.4 (2010). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wsbm.75, pp. 438–459. ISSN: 1939-005X. DOI: 10.1002/wsbm.75. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/wsbm.75 (visited on 05/05/2024).

[14] Sina Anvari et al. "Computational Models and Simulations of Cancer Metastasis". en. In: *Archives of Computational Methods in Engineering* 28.7 (Dec. 2021), pp. 4837–4859. ISSN: 1886-1784. DOI: 10.1007/s11831-021-09554-1. URL: https://doi.org/10.1007/s11831-021-09554-1 (visited on 05/05/2024).

[15] John Metzcar et al. “A Review of Cell-Based Computational Modeling in Cancer Biology”. In: *JCO Clinical Cancer Informatics* 3 (Dec. 2019). Publisher: Wolters Kluwer, pp. 1–13. DOI: 10.1200/CCI.18.00069. URL: https://ascopubs.org/doi/10.1200/CCI.18.00069 (visited on 05/17/2024).

[16] Fleur Jeanquartier et al. “Machine Learning for In Silico Modeling of Tumor Growth”. en. In: *Machine Learning for Health Informatics: State-of-the-Art and Future Challenges*. Ed. by Andreas Holzinger. Cham: Springer International Publishing, 2016, pp. 415–434. ISBN: 978-3-319-50478-0. DOI: 10.1007/978-3-319-50478-0_21. URL: https://doi.org/10.1007/978-3-319-50478-0_21 (visited on 05/05/2024).

[17] Olga Maria Dona Lemus et al. “Adaptive Radiotherapy: Next-Generation Radiotherapy”. en. In: *Cancers* 16.6 (Jan. 2024). Number: 6 Publisher: Multidisciplinary Digital Publishing Institute, p. 1206. ISSN: 2072-6694. DOI: 10.3390/cancers16061206. URL: https://www.mdpi.com/2072-6694/16/6/1206 (visited on 08/05/2024).

[18] Jan-Jakob Sonke, Marianne Aznar, and Coen Rasch. “Adaptive Radiotherapy for Anatomical Changes”. In: *Seminars in Radiation Oncology*. Adaptive Radiotherapy and Automation 29.3 (July 2019), pp. 245–257. ISSN: 1053-4296. DOI: 10.1016/j.semradonc.2019.02.007. URL: https://www.sciencedirect.com/science/article/pii/S1053429619300165 (visited on 08/05/2024).

[19] Di Yan et al. “Adaptive radiation therapy”. en. In: *Physics in Medicine & Biology* 42.1 (Jan. 1997), p. 123. ISSN: 0031-9155. DOI: 10.1088/0031-9155/42/1/008. URL: https://dx.doi.org/10.1088/0031-9155/42/1/008 (visited on 08/14/2024).

[20] Carri K. Glide-Hurst et al. “Adaptive Radiation Therapy (ART) Strategies and Technical Considerations: A State of the ART Review From NRG Oncology”. In: *International Journal of Radiation Oncology*Biology*Physics* 109.4 (Mar. 2021), pp. 1054–1075. ISSN: 0360-3016. DOI: 10.1016/j.ijrobp.2020.10.021. URL: https://www.sciencedirect.com/science/article/pii/S0360301620344096 (visited on 08/14/2024).

[21] Martin Burger. “Parameter identification”. In: *Lecture Notes* (2005).

[22] Paul Meißner, Tom Hoppe, and Thomas Vietor. “Comparative Study of Various Neural Network Types for Direct Inverse Material Parameter Identification in Numerical Simulations”. en. In: *Applied Sciences* 12.24 (Jan. 2022). Number: 24 Publisher: Multidisciplinary Digital Publishing Institute, p. 12793. ISSN: 2076-3417. DOI: 10.3390/app122412793. URL: https://www.mdpi.com/2076-3417/12/24/12793 (visited on 05/23/2024).

[23] Bence Czako, Daniel Andras Drexler, and Levente Kovacs. “Time-Varying Parameter Identification of a Tumor Growth Model Using Moving Horizon Estimation”. In: *2022 IEEE 26th International Conference on Intelligent Engineering Systems (INES)*. Georgioupolis Chania, Greece: IEEE, Aug. 2022, pp. 000073–000078. ISBN: 978-1-66549-209-6. DOI: 10.1109/INES56734.2022.9922626. URL: https://ieeexplore.ieee.org/document/9922626/ (visited on 05/22/2024).

[24] Ziheng Sun, Liping Di, and Hui Fang. “Using long short-term memory recurrent neural network in land cover classification on Landsat and Cropland data layer time series”. In: *International Journal of Remote Sensing* 40.2 (Jan. 2019). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01431161.2018.1516313, pp. 593–614. ISSN: 0143-1161. DOI: 10.1080/01431161.2018.1516313. URL: https://doi.org/10.1080/01431161.2018.1516313 (visited on 05/23/2024).

[25] Marc RuBwurm and Marco Korner. “Temporal Vegetation Modelling Using Long Short-Term Memory Networks for Crop Identification from Medium-Resolution Multi-spectral Satellite Images”. en. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Honolulu, HI, USA: IEEE, July 2017, pp. 1496–1504. ISBN: 978-1-5386-0733-6. DOI: 10.1109/CVPRW.2017.193. URL: http://ieeexplore.ieee.org/document/8014927/ (visited on 05/22/2024).

[26] Lingyan Que et al. “A Robust Line Parameter Identification Method Based on LSTM and Modified SCADA Data”. In: *2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2)*. Wuhan, China: IEEE, Oct. 2020, pp. 2981–2986. ISBN: 978-1-72819-606-0. DOI: 10.1109/EI250167.2020.9346604. URL: https://ieeexplore.ieee.org/document/9346604/ (visited on 05/22/2024).

[27] Charlotte Pelletier, Geoffrey I. Webb, and François Petitjean. “Temporal Convolutional Neural Network for the Classification of Satellite Image Time Series”. en. In: *Remote Sensing* 11.5 (Jan. 2019). Number: 5 Publisher: Multidisciplinary Digital Publishing Institute, p. 523. ISSN: 2072-4292. DOI: 10.3390/rs11050523. URL: https://www.mdpi.com/2072-4292/11/5/523 (visited on 05/22/2024).

[28] Artemios-Anargyros Semenoglou, Evangelos Spiliotis, and Vassilios Assimakopoulos. “Image-based time series forecasting: A deep convolutional neural network approach”. In: *Neural Networks* 157 (Jan. 2023), pp. 39–53. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2022.10.006. URL: https://www.sciencedirect.com/science/article/pii/S0893608022003902 (visited on 05/28/2024).

[29] Zuxuan Wu et al. "Modeling Spatial-Temporal Clues in a Hybrid Deep Learning Framework for Video Classification". en. In: *Proceedings of the 23rd ACM international conference on Multimedia*. Brisbane Australia: ACM, Oct. 2015, pp. 461–470. ISBN: 978-1-4503-3459-4. DOI: 10.1145/2733373.2806222. URL: https://dl.acm.org/doi/10.1145/2733373.2806222 (visited on 05/22/2024).

[30] Zhenfei Guo et al. "CPINet: Parameter identification of path-dependent constitutive model with automatic denoising based on CNN-LSTM". In: *European Journal of Mechanics - A/Solids* 90 (Nov. 2021), p. 104327. ISSN: 0997-7538. DOI: 10.1016/j.euromechsol.2021.104327. URL: https://www.sciencedirect.com/science/article/pii/S099775382100098X (visited on 04/08/2024).

[31] Carlos Iturrino Garcia et al. "A Comparison of Power Quality Disturbance Detection and Classification Methods Using CNN, LSTM and CNN-LSTM". en. In: *Applied Sciences* 10.19 (Jan. 2020). Number: 19 Publisher: Multidisciplinary Digital Publishing Institute, p. 6755. ISSN: 2076-3417. DOI: 10.3390/app10196755. URL: https://www.mdpi.com/2076-3417/10/19/6755 (visited on 05/27/2024).

[32] Johann Rudi, Julie Bessac, and Amanda Lenzi. "Parameter Estimation with Dense and Convolutional Neural Networks Applied to the FitzHugh–Nagumo ODE". en. In: ().

[33] Geoffrey M. Cooper. "The Eukaryotic Cell Cycle". en. In: *The Cell: A Molecular Approach. 2nd edition*. Sinauer Associates, 2000. URL: https://www.ncbi.nlm.nih.gov/books/NBK9876/ (visited on 05/19/2024).

[34] Michel Verleysen, John Lee. *LELEC2870 - Machine learning : regression, deep networks and dimensionality reduction*. English. Lectures. Université catholique de Louvain.

[35] Saerens Marco. *LINFO2275 - Data Mining and Decision Making*. English. Lectures. Université catholique de Louvain.

[36] Dupont Pierre. *LINFO2262 Machine Learning : classification and evaluation*. English. Lectures. Université catholique de Louvain.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**