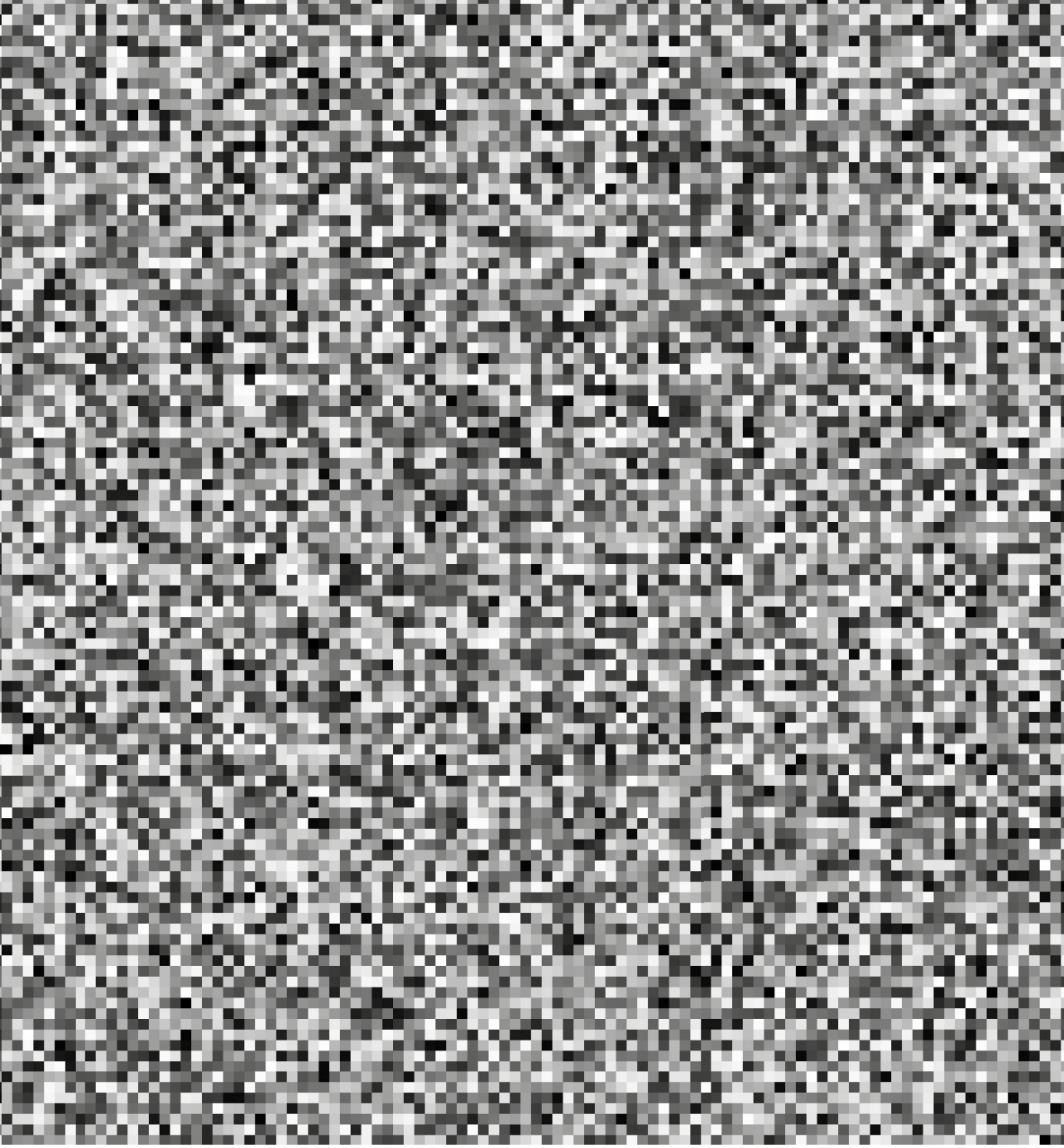
# Distributed learning and prediction modelling in radiation oncology



# Distributed learning and prediction modelling in radiation oncology

Timo M. Deist

## Colofon

Layout and cover design: Frank J.W.M. Dankers, Djoya D.N. Hattu
Printing: Proefschriftmaken, Proefschriftmaken.nl
ISBN: 978-90-829801-6-5

# Distributed learning and prediction modelling in radiation oncology

Dissertation

to obtain the degree of Doctor at the Maastricht University,
on the authority of the Rector Magnificus
Prof. dr. Rianne M. Letschert
in accordance with the decision of the Board of Deans,
to be defended in public on Friday 5 April 2019 at 16.00h

by

Timo Matthias Deist

| | |
|---|---|
| Supervisors | Prof. dr. P. Lambin<br>Prof. dr. A. Dekker |
| Co-supervisor | Dr. A. Jochems |
| Assessment Committee | Prof. dr. Frank Verhaegen (chair)<br>Prof. dr. Jos CS Kleinjans<br>Prof. dr. Benoît Macq (UC Louvain)<br>Prof. dr. Dirk De Ruysscher<br>Dr. Xander Verbeek (IKNL) |

# Table of contents

# Chapter 1

## Introduction

Research on cancer is an industry fueled by public funding (estimated 11 billion US dollar, 2004/2005[1]), private investment (estimated 3 billion US dollar by the top 24 pharmaceutical companies, 2004/2005[1]), and the scientific vigor of the global research community.
Another essential resource for cancer research is clinical data from healthcare providers: data describing oncology patients, their treatments, and therapy results. This data highlights the successes of cancer research and where treatment improvements are necessary. It is a measurement directly at the focal point: the treatment of a cancer patient—for which the cancer research industry has been built and where all research projects coalesce. If enough clinical data is available, statistical data analysis can not only evaluate the quality of cancer treatments but also transform clinical data into information to drive future treatment decisions: based on data from previous patients, which treatment should a new patient receive to maximize their chance of survival? Which treatment will minimize the chance of negative side effects? Clinical data thus becomes an input for cancer research.
When the clinical data and treatment decision become complex, statistical analysts use abstract computational models to solve the research problem. At this point, the analyst delegates a part of their work to a machine which analyzes (*learns*) from this clinical data to find the correct answer. Such computational models form the basis for *machine learning*, an emerging subfield of statistics.

## Machine learning

Machine learning is a statistical process in which computational algorithms identify patterns in datasets. Datasets, in our context of clinical data analysis, consist of characteristics describing the disease and treatment of multiple patients. The patterns identified by the algorithm allow further describing a patient, e.g., categorize patients into subgroups (*clustering*, an *unsupervised* machine learning technique) or predict the probability of survival after treatment (*prediction modelling*, a *supervised* machine learning technique). The machine learning algorithms treated in this thesis are supervised learning algorithms for prediction modelling (with exception of the nearest neighbor algorithms used in chapter 5). Many machine learning algorithms have been developed in the last decades and it is not always clear which algorithm is the most suitable for a given task. The second part of this thesis concerns the comparison of existing algorithms for prediction modelling in radiation oncology and the development of novel algorithms.
Machine learning is used successfully for many tasks outside medical research, e.g., automated language translation or image recognition, and has become increasingly popular in cancer research. Cancer research publications involving machine learning quintupled from 2010 to 2017[1].

[1] The relative number of publications listed on PubMed involving cancer or oncology and machine learning versus publications involving only cancer or oncology. Pubmed search (20.09.2018): ((cancer) OR oncology) AND (machine learning) versus ((cancer) OR oncology).

## Access to data and distributed learning

For a machine to learn and provide accurate answers to a cancer research question, e.g., which treatment provides the highest survival probability for a given patient, it needs to have access to large amounts of clinical data. Access to clinical data is difficult for multiple reasons[2], for example:

- regulations protecting patient privacy: medical details should not become public;
- technical barriers: how to transfer large clinical data volumes?;
- lacking data standardization: information is stored in incompatible formats;
- competing interests: institutes do not share clinical data as it is a valuable resource for research;
- PR risks: sharing therapy results allows comparing performance across healthcare providers.

To overcome the regulatory and technical barriers, the concept of distributed machine learning can be employed. Instead of collecting all clinical data in a central database, which risks patient privacy violations, and then applying the machine learning algorithm, distributed learning uses a different approach: the data remains with the healthcare provider and the algorithm is sent to the data as proposed by Gaye et al. (2014)[3]. Figure 1 illustrates the distributed learning concept.

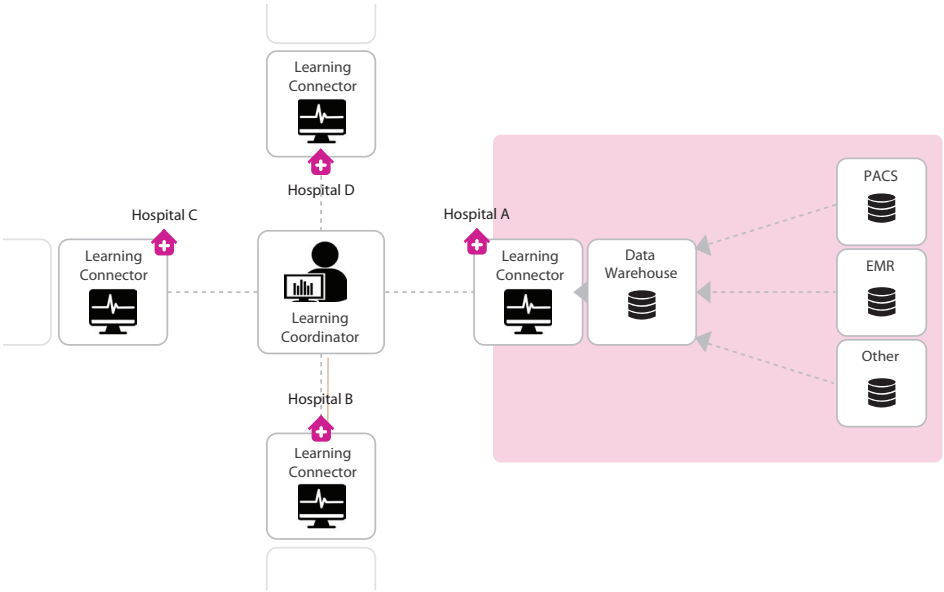



**Figure 1.** An example distributed learning infrastructure with four participating hospitals. All hospitals installed a Learning Connector that communicates with the Learning Coordinator server outside the hospitals. The Learning Connector receives machine learning algorithms and applies them on the hospital's data (extracted from various local databases), the learning results are sent back to the Learning Coordinator. Adapted from Lambin et al. (2017)[4].

In this way, patient-specific information does not leave the healthcare provider because the machine only learns and stores abstract concepts which are equivalent to aggregated patient information. Furthermore, no large data volumes need to be exchanged except for the moderately sized machine learning algorithm. However, developing a distributed learning infrastructure is a technical challenge in itself. Furthermore, already established machine learning algorithms need to be redesigned to function in a distributed learning setting. We therefore distinguish between two types of learning algorithms

- *centralized* learning algorithms: learning on data in a single database;
- *distributed* learning algorithms: learning on data spread over multiple databases.

The implementation of a distributed learning infrastructure and algorithms forms the first part of this thesis. Centralized learning algorithms are studied in the second part of this thesis.



## Radiation oncology

Access to clinical data is an issue for medical research in general and distributed learning potentially offers solutions for all medical specializations. This thesis focusses on oncology and radiation oncology (or *radiotherapy*) in particular. Radiotherapy, surgery, and chemotherapy are the most frequently used cancer treatments. In radiotherapy, the tumor is irradiated either by an external radiation beam or by a radiation source that is surgically inserted in close proximity of the tumor (*brachytherapy*). The radiation causes DNA damage in the tumor and surrounding tissues. The radiation treatment is often repeated multiple times. DNA damage in the surrounding tissues can cause negative side effects, for example, swallowing problems (dysphagia) due to irradiation of the esophagus during external beam lung radiotherapy. The prediction of side effects and patient survival (i.e. treatment outcomes) are the main goal of machine learning algorithms discussed in this thesis.

## This thesis

This thesis has two main subjects:

- the development and implementation of a distributed learning infrastructure across international radiotherapy institutes (Chapters 2-3);
- studies of centralized machine learning algorithms (Chapters 4-5).

The latter comprises empirical analyses of existing classification algorithms for treatment outcome prediction and the development of novel machine learning algorithms. See Table 1 for an overview.

- **Chapter 2** outlines the distributed learning concept and initial results. We present the implementation of a distributed support vector machine algorithm described by Boyd et al. (2010)[5] and its application in our first distributed learning project in four institutes spanning three countries (Belgium/Germany/The Netherlands). This chapter forms one of our multiple early studies[6,7] on the distributed learning infrastructure.
- **Chapter 3** presents results from a large-scale follow-up study across eight institutes in five countries (England/Italy/The Netherlands/The People's Republic of China/Wales). It introduces a new implementation of a distributed logistic regression algorithm (Boyd et al. (2010)[5]) applied on survival and cancer staging data of more than 20 000 non-small cell lung cancer patients.
- **Chapter 4** presents results of an empirical comparison of binary classification algorithms for modelling treatment outcomes in radiotherapy on 12 datasets.

- **Chapter 5** introduces novel kernelized classification and regression algorithms which allow exploiting (bioinformatic) simulation models in machine learning algorithms. We study these algorithms in four exemplary cases from bioinformatics and network flow optimization.
- **Chapter 6** discusses challenges for the acceptance and sustainability of distributed learning infrastructures followed by an appraisal of machine learning in radiotherapy research.

**Table 1.** Thesis structure.

| **Introduction** | **Original research** | | | | **Discussion** |
|---|---|---|---|---|---|
| **Chapter 1** | **Chapter 2** | **Chapter 3** | **Chapter 4** | **Chapter 5** | **Chapter 6** |
| | Distributed learning | | Centralized learning | | |
| | Infrastructure development | | | | |
| | Algorithm implementation | | Algorithm comparison | Algorithm development | |

## References


1. Eckhouse, S., Lewison, G. & Sullivan, R. Trends in the global funding and activity of cancer research. *Molecular Oncology* **2**, 20–32 (2008).
2. Sullivan, R. *et al*. Delivering affordable cancer care in high-income countries. *The Lancet Oncology* **12**, 933–980 (2011).
3. Gaye, A. *et al*. DataSHIELD: taking the analysis to the data, not the data to the analysis. *Int J Epidemiol* **43**, 1929–1944 (2014).
4. Lambin, P. *et al*. Radiomics: the bridge between medical imaging and personalized medicine. *Nature Reviews Clinical Oncology* **14**, 749–762 (2017).
5. Boyd, S. *et al.* Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends® in Machine Learning* **3**, 1–122 (2010).
6. Jochems, A. *et al*. Distributed learning: Developing a predictive model based on data from multiple hospitals without data leaving the hospital – A real life proof of concept. *Radiotherapy and Oncology* **121**, 459–467 (2016).
7. Jochems, A. *et al*. Developing and Validating a Survival Prediction Model for NSCLC Patients Through Distributed Learning Across 3 Countries. *International Journal of Radiation Oncology*Biology*Physics* **99**, 344–352 (2017).

# Distributed learning

# Chapter 2

## Infrastructure and distributed learning methodology for privacy-preserving multi-centric rapid learning health care: euroCAT

Timo M. Deist, A. Jochems, Johan van Soest, Georgi Nalbantov, Cary Oberije, Seán Walsh, Michael Eble, Paul Bulens, Philippe Coucke, Wim Dries, Andre Dekker, Philippe Lambin

***Adapted from*** *Deist, Timo M., et al. "Infrastructure and distributed learning methodology for privacy-preserving multi-centric rapid learning health care: euroCAT." Clinical and translational radiation oncology 4 (2017): 24-31.*

**Abstract**

Machine learning applications for personalized medicine are highly dependent on access to sufficient data. For personalized radiation oncology, datasets representing the variation in the entire cancer patient population need to be acquired and used to learn prediction models. Ethical and legal boundaries to ensure data privacy hamper collaboration between research institutes. We hypothesize that data sharing is possible without identifiable patient data leaving the radiation clinics and that building machine learning applications on distributed datasets is feasible. We developed and implemented an IT infrastructure in five radiation clinics across three countries (Belgium, Germany, and The Netherlands). We present here a proof-of-principle for future 'big data' infrastructures and distributed learning studies. Lung cancer patient data was collected in all five locations and stored in local databases. Exemplary support vector machine (SVM) models were learned using the Alternating Direction Method of Multipliers (ADMM) from the distributed databases to predict postradiotherapy dyspnea grade ≥ 2. The discriminative performance was assessed by the area under the curve (AUC) in a five-fold cross-validation (learning on four sites and validating on the fifth). The performance of the distributed learning algorithm was compared to centralized learning where datasets of all institutes are jointly analyzed. The euroCAT infrastructure has been successfully implemented in five radiation clinics across three countries. SVM models can be learned on data distributed over all five clinics. Furthermore, the infrastructure provides a general framework to execute learning algorithms on distributed data. The ongoing expansion of the euroCAT network will facilitate machine learning in radiation oncology. The resulting access to larger datasets with sufficient variation will pave the way for generalizable prediction models and personalized medicine.

## Introduction

Medical research revolves around accumulation and analysis of (patient) data. Collecting sufficient quantities of data to explain a phenomenon is arguably a major impediment to scientific progress in a technology-driven discipline such as radiation oncology. This obstacle becomes even more eminent in light of the recent adoption of machine learning[1] to foster the goal of personalized medicine: machine learning algorithms require access to large databases with sufficient variation in the collected data to answer complex research questions. Single institutes struggle to collect the necessary data volumes with sufficient diversity to learn from. Furthermore, data collected in radiation oncology is influenced and biased by technological (e.g., vendor-specific properties[2]), human (e.g., local patient characteristics, physician's opinions[3]), as well as organizational (e.g., treatment guidelines) factors which can change rapidly.



Research questions in such contexts may remain unanswerable by isolated data collection efforts: the data may be too biased or simply lack the necessary variation to successfully model relationships between the collected variables. Data homogeneity may not only be an issue for single institutes but nationwide due to national treatment guidelines[4]. Hence, generalizable machine learning models to answer these research questions should be created by incorporating data from multiple institutes in a continuous manner (i.e. rapid learning health care[5]). Systematic data sharing among research institutes will become an indispensable means for personalized medicine to thrive in radiation oncology. At present, data sharing is characterized by one-off exchanges of datasets with limited standardization of data collection and data characterization. Further, data sharing is impeded by each institute's legal and ethical concern to protect their patients' privacy rights. In this study, we present euroCAT, an IT infrastructure for systematic data sharing among research institutes. A video summary is available here: **https://youtu.be/ZDJFOxpwqEA**. The hypotheses of the study are

1. Data sharing for machine learning is possible without identifiable patient data leaving an institute's IT systems. Thus, the institutes remain in control of their data, preserve data privacy, and thereby overcome legal and ethical issues common to other forms of data exchanges.
2. Running machine learning applications on these data is feasible and, given the appropriate methodology, the resulting models only minimally differ from centrally learned models, which makes efforts to centralize data largely unnecessary. As an example, support vector machines (SVM) predicting severe dyspnea after radiotherapy (henceforth simply called dyspnea) are learned from the data provided in five institutes.

The aim of the study was to deploy the euroCAT system in five partner institutions within three European countries (Belgium, Germany, and The Netherlands) and in four languages (Dutch, English, French, and German) and test the above hypotheses. euroCAT focusses on multi-centric machine learning in radiation oncology, similar work to implement privacy-preserving data analysis exists, e.g., for Genome-Wide Association Studies (GWAS)[6]. Constable et al. (2015) concisely discuss existing literature for distributed learning and the accompanying risks. A web service for distributed logistic regression analysis is presented by Jiang et al. (2013)[7] to facilitate collaborative regression analysis.

## Material & methods

### *euroCAT infrastructure*

Institutes within the euroCAT network (a site) dedicate a server within their IT infrastructure that hosts the local databases and local learning connector (Varian Medical Systems, Palo Alto, USA). The global learning environment (Varian Learning Portal) spans the sites, and connects a central server (the master) outside the sites' IT infrastructure to the learning connectors inside the sites. Master and sites communicate via file-based, asynchronous messaging. The user interacts with the learning environment via a web browser-based interface in which s/he can upload learning applications (MATLAB, MathWorks, Natick, MA, USA) and can initiate machine learning runs. Every learning application consists of two parts, one site algorithm which runs inside the sites' infrastructure and interacts with the learning connector and one master algorithm which runs in the global learning environment and can send and receive messages to and from the site algorithms.

### *Data*

Each participating center (Aachen (Germany), Eindhoven (The Netherlands), Hasselt (Belgium), Liège (Belgium), and Maastricht (The Netherlands)) was asked to retrospectively select at least 50 patients which fulfilled the inclusion criteria (non-small cell lung cancer, high-dose radiotherapy, no surgical treatment). The centers were provided with an overview of variables that were needed for the study. Initially, survival outcome, dysphagia outcome, and dyspnea outcome were scored. For this proof-of-principle paper, we only used the dyspnea outcome. The data was stored in a spreadsheet. An euroCAT researcher visited each center and manually checked 20% of the collected data for inconsistencies/mistakes. Post-treatment dyspnea was recorded for 268 patients. Given availability in the databases, three features were manually selected to construct an exemplary prediction model for post-treatment severe dyspnea: lung function tests (FEV1 (in %), forced expiratory volume in 1 s, in %, adjusted for age and gender), cardiac comorbidity (non-hypertension cardiac disorder at baseline, for which treatment at a cardiology department has been given), and timing of chemotherapy. Severe dyspnea was defined as ≥ Grade 2 dyspnea after treatment. The variables are listed in Table 1.

From the spreadsheets, data was extracted using an open source data warehousing tool (Pentaho) and stored in an open-source database (PostgreSQL). From this database, data elements were mapped to the Semantic Web data model (Resource Description Framework, RDF) using an open source tool (D2RQ) and stored in an open-source RDF store (Sesame, Eclipse RDF4J). During mapping to RDF the data elements were coded using Uniform Resource Identifiers (URIs) which are defined in a domain ontology (Radiation Oncology Ontology) and reference ontologies (NCI Thesaurus, Unit Ontology) in the Web Ontology Language (OWL, available on the Bioportal[8]). The learning connector uses the Semantic Web query language SPARQL to query data from the RDF store[9] and can parse that data to the site learning algorithm.

**Table 1.** Overview of patient characteristics per hospital.

| | Maastricht | | Eindhoven | | Hasselt | | Liège | | Aachen | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Variable** | **Count** | **%** | **Count** | **%** | **Count** | **%** | **Count** | **%** | **Count** | **%** |
| **Post-RT Dyspnea** | | | | | | | | | | |
| < 2 | 89 | 72% | 50 | 89% | 8 | 57% | 20 | 61% | 36 | 86% |
| ≥ 2 | 34 | 28% | 6 | 11% | 6 | 43% | 13 | 39% | 6 | 14% |
| Missing | 0 | 0% | 0 | 0% | 0 | 0% | 0 | 0% | 0 | 0% |
| **Cardiac Comorbidity** | | | | | | | | | | |
| No | 90 | 73% | 44 | 79% | 2 | 14% | 27 | 82% | 24 | 57% |
| Yes | 33 | 27% | 12 | 21% | 3 | 21% | 6 | 18% | 12 | 29% |
| Missing | 0 | 0% | 0 | 0% | 9 | 64% | 0 | 0% | 6 | 14% |
| **Chemotherapy Timing** | | | | | | | | | | |
| None | 16 | 13% | 5 | 9% | 3 | 21% | 0 | 0% | 2 | 5% |
| Sequential | 22 | 18% | 24 | 43% | 2 | 14% | 2 | 6% | 4 | 10% |
| Concurrent | 85 | 69% | 27 | 48% | 8 | 57% | 31 | 94% | 33 | 79% |
| Missing | 0 | 0% | 0 | 0% | 1 | 7% | 0 | 0% | 3 | 7% |
| **FEV1 (in %)** | | | | | | | | | | |
| Mean & Standard Dev | 78 | 21 | 80 | 25 | 80 | 25 | 72 | 23 | 66 | 19 |
| Missing Count & Percentage | 0 | 0% | 20 | 36% | 2 | 14% | 0 | 0% | 20 | 48% |

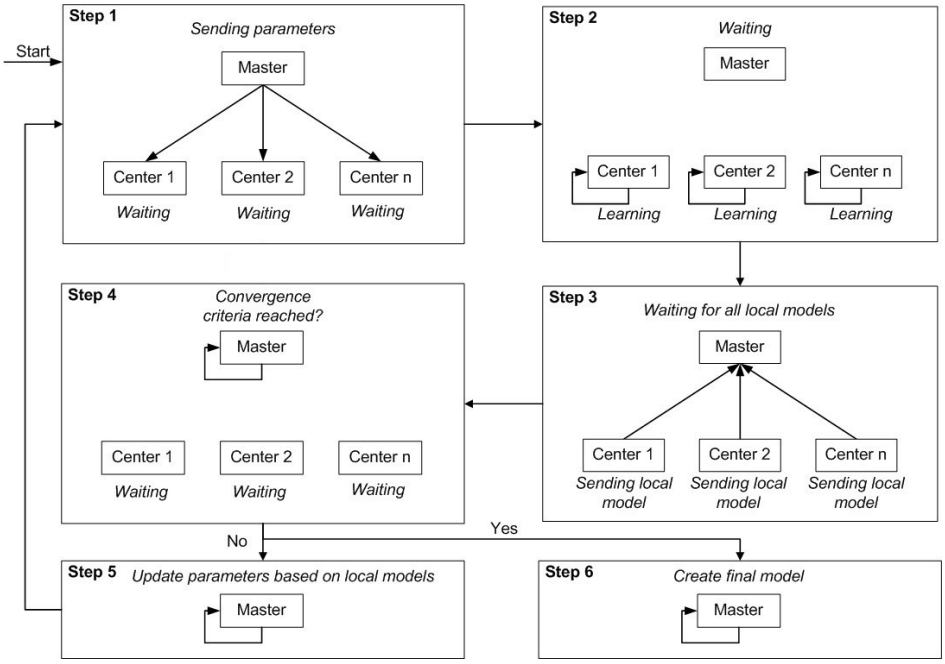



**Figure 1.** Distributed learning flow in euroCAT.

*Distributed learning*

The process of carrying out distributed learning inside euroCAT is presented schematically in Fig. 1. At each iteration, the data stored at different sites is processed simultaneously and separately. Updated model parameters are then sent from each site to the master. At the master, an algorithm compares the model parameters and updates them further. The algorithm also checks whether the learning process has converged sufficiently (according to pre-set convergence criteria). If the convergence criteria are not yet met, the master sends the parameters back to each of the sites. Once the sites receive updated parameters, they are used as a starting point for adjusting the model parameters further (given the local data) once again. This completes one iteration cycle. The learning iterations continue until the convergence criteria are satisfied. Using this infrastructure allows models to use data for learning without transferring these data across the network. The learned model is a support vector machine (SVM) classifier, solved with the Alternating Direction Method of Multipliers (ADMM) method[10].

*Support vector machines (SVM)*
*& the Alternating Direction Method of Multipliers (ADMM)*

A support vector machine determines two parallel hyperplanes, forming a 'border' which separates the feature space into two large regions and a margin between the planes. Each dimension of this feature space represents one patient feature (e.g., FEV1 (in %) or cardiac comorbidity) and each patient is represented by one point in this space. For simple problems, the intention would be to identify hyperplanes that separate all patients with dyspnea from the group of patients without dyspnea. This is not possible in most cases, therefore the objective becomes to find hyperplanes such that

- most of the dyspneic patients are on one and non-dyspneic patients are on the other side;
- if there is a patient on the 'wrong' side of the border, the distance to the border is as small as possible;
- the border between the groups of dyspneic and non-dyspneic patients is as large as possible.

The optimal hyperplanes ($dw+b=1$ and $dw+b=-1$, where $d$ are the features of a patient) are determined by a vector of coefficients $(w,b)$ that minimizes a cost function under a set of constraints (see Appendix A for details). Boyd et al. (2011)[10] discuss a distributed formulation of a support vector machine using the Alternating Direction Method of Multipliers (ADMM) and provide MATLAB code[11]. The ADMM algorithm gained popularity in the machine learning community as it allows to split up large datasets into smaller portions and distribute the analysis over multiple machines. In our multi-centric learning context, the same property is exploited to overcome the restriction that data may not be centralized. ADMM requires a multitude of iterations in which estimates for $(w,b)$ are refined using each site's data. See Appendix A for a more detailed description of SVMs and ADMM.

*Learning & validation*

For details on additional data processing steps, parametrization of the ADMM algorithm, and the code used to execute the algorithm, we refer to Appendix B.

To display the capabilities of the distributed learning network, support vector machines are once trained on all sites and once trained and validated in a cross-validation design: the SVM is fitted using data from four sites and validated on the remaining site. This process is repeated four times with validation on another site. The average values for training and validation constitute the cross-validation result.

The models' performance is measured in terms of discriminative performance expressed as the area under the curve (AUC) of the Receiver Operating Characteristic curve (ROC).

To demonstrate the validity of the distributed learning approach with respect to a centralized learning algorithm, we compare the ADMM results to solutions from a centralized SVM optimizer. To this end, we centralize the data from all sites and solve the SVM optimization problem (Eqs. (1)–(3), Appendix A). Missing value imputation is still done per site to ensure comparability of centralized and distributed results. For this demonstration, the distributed algorithm is run in a local simulation environment.

## Results & discussion

The results for learning and validation on all sites and the 5-fold cross-validation can be found in Table 2. The discriminative performance in the cross-validation is modest with a validation AUC of 0.66 but stable across training (0.62) and validation (0.66). Training AUCs are stable across folds (0.60–0.64) while inter-fold validation AUCs vary considerably (0.57–0.77). Published models[12,13] show similar discriminative performance. The sole purpose of the presented SVM models is to display the infrastructure's functionality and it is advised not to use these models in a clinical setting.

The coefficients of the SVM trained in the euroCAT network and in centralized learning can be found in Table 4. The individual run time of the 6 learning runs in the current euroCAT network was approximately 2 h or less with an iteration count between 300 and 500. Fig. 2 illustrates the convergence of the ADMM results to the centralized optimization results for all six learning runs. The iteration number is listed on the x-axis, the norm of the difference between ADMM and centralized results is shown on the y-axis. The algorithm was run for $10^4$ iterations and the iterations in which the internal convergence criteria are met in the euroCAT network are indicated by vertical lines. In all six cases, the solution approaches the centralized solution non-monotonically until the convergence criteria are met and the ADMM algorithm stops. The ADMM-based SVMs do not completely coincide with centralized models (see Table 4) as the convergence criteria were relaxed to accommodate for the relatively long network communication time in each iteration. A centralized learning algorithm determines SVM coefficients in less time as there is no network communication. Thus, when using ADMM-based distributed learning (or other distributed learning methods with repeating master-site communication), one faces a trade-off between solution precision and computation time. While the network communication time will surely force large-scale simulation studies to be maximally parallelized (to minimize the impact of network communication), the impact on prediction model development and performance is expected to be limited: the impact on AUC-based discriminative performance is small for the exemplary SVM models (compare Tables 2 and 3) and can be further reduced with stricter convergence criteria (see Fig. 2). Viewed differently, 'early stopping' is employed in machine learning as a

regularization technique to avoid overfitting[14]. Models that suffer from overfitting explain the training data but fail to correctly predict outcomes in other datasets. Therefore, the trade-off between solution precision and computation time should not harm the goal of developing robust machine learning models for personalized medicine.

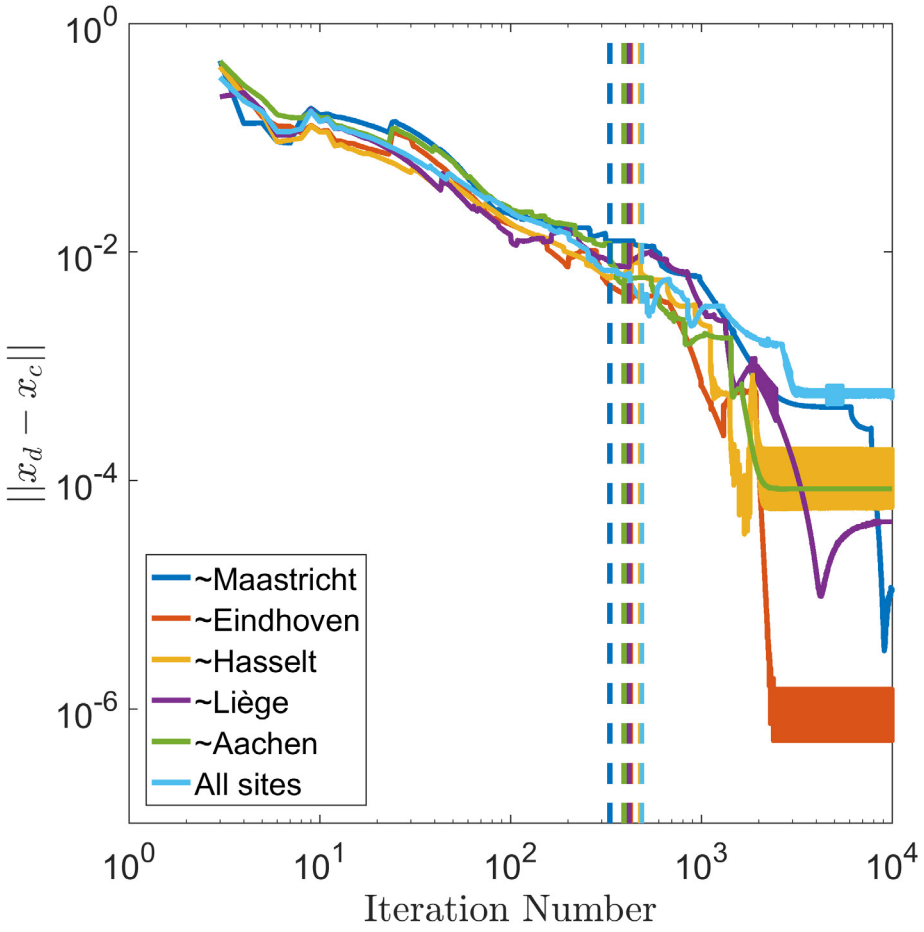



**Figure 2.** Convergence graphs of distributed ADMM solutions $x_d$ to centralized solutions $x_c$ for $10^4$ iterations. Vertical lines indicate the iterations in which internal convergence criteria were met in the euroCAT network. The data was created in local simulations. '~' indicates 'Trained on all sites except'.

**Table 2.** Discrimination performance (AUC) obtained by learning an SVM on all sites and in a 5-fold CV in distributed learning (ADMM, following the formulation shown in Eqs. (4)–(7), Appendix A).

| | | | | | | | CV |
|---|---|---|---|---|---|---|---|
| **Train on** | **All** | **All except Maastricht** | **All except Eindhoven** | **All except Hasselt** | **All except Liège** | **All except Aachen** | |
| **Validate on** | | **Maastricht** | **Eindhoven** | **Hasselt** | **Liège** | **Aachen** | |
| **Training AUC** | 0.63 | 0.61 | 0.60 | 0.64 | 0.62 | 0.64 | 0.62 |
| **Validation AUC** | | 0.58 | 0.77 | 0.57 | 0.72 | 0.64 | 0.66 |

**Table 3**. Discrimination performance (AUC) obtained by learning an SVM on all sites and in a 5-fold CV in centralized learning (solving the optimization problem shown in equations 1-3, Appendix A).

| | | | | | | | CV |
|---|---|---|---|---|---|---|---|
| **Train on** | **All** | **All except Maastricht** | **All except Eindhoven** | **All except Hasselt** | **All except Liège** | **All except Aachen** | |
| **Validate on** | | **Maastricht** | **Eindhoven** | **Hasselt** | **Liège** | **Aachen** | |
| **Training AUC** | 0.63 | 0.61 | 0.60 | 0.63 | 0.61 | 0.64 | 0.62 |
| **Validation AUC** | | 0.58 | 0.77 | 0.59 | 0.72 | 0.64 | 0.66 |

**Table 4.** SVM coefficients ($w$,$b$) learned by distributed and centralized learning.

| **Trained on** | | $w_1$ | $w_2$ | $w_3$ | $w_4$ | $b$ |
|---|---|---|---|---|---|---|
| **All** | **Distributed** | 0.01 | -0.32 | -0.20 | -0.25 | -0.55 |
| | **Centralized** | 0.01 | -0.31 | -0.20 | -0.25 | -0.55 |
| **All except Maastricht** | **Distributed** | -0.03 | -0.31 | -0.20 | -0.29 | -0.51 |
| | **Centralized** | -0.02 | -0.31 | -0.20 | -0.29 | -0.51 |
| **All except Eindhoven** | **Distributed** | 0.01 | -0.28 | -0.06 | -0.33 | -0.48 |
| | **Centralized** | 0.02 | -0.28 | -0.06 | -0.33 | -0.48 |
| **All except Hasselt** | **Distributed** | 0.00 | -0.32 | -0.20 | -0.26 | -0.55 |
| | **Centralized** | 0.00 | -0.31 | -0.20 | -0.26 | -0.55 |
| **All except Liège** | **Distributed** | 0.00 | -0.31 | -0.20 | -0.25 | -0.55 |
| | **Centralized** | -0.01 | -0.31 | -0.20 | -0.26 | -0.55 |
| **All except Aachen** | **Distributed** | 0.00 | -0.34 | -0.19 | -0.24 | -0.53 |
| | **Centralized** | 0.00 | -0.34 | -0.19 | -0.24 | -0.53 |

A challenge of distributed learning is that the user is not able to inspect the data which is used as input for the machine learning applications. S/he must rely on summary statistics to ascertain that the data is in the desired format. This obstacle can be overcome by collaboration between users from the respective institutes and strictly following the agreed data collection and storage protocols. An euroCAT umbrella protocol[15] was provided to the participating institutes to guide future lung data collection. Protocols for other diseases are also available: for a data sharing project between MAASTRO Clinic and the Sacred Heart University Hospital (Rome) on rectal cancer, a corresponding umbrella protocol was developed[16].

Systematic data sharing not only requires an IT infrastructure, as developed in this study, but it also depends on systematic data collection in routine clinical care. It has been argued that data from routine care is a valuable source of information to improve the standard of care[5,17]. However, this data is often not treated as such. Consequently, data collection and standardization have the potential to be improved as also observed in this study.

Even though routine clinical care might become a cornucopia of clinical data, this data needs to be handled with care: McGale et al. (2016)[18] show that conclusions from routine clinical care data may contradict findings from randomized clinical trials. Routine care data is subject to many biases contrary to data from carefully designed trials. The conclusion should not be to discard routine care data altogether but rather to develop means to profit from this data: i.e., develop appropriate methodology, e.g., extensive correction for confounders[19], and to expand standardized data collection to capture all data necessary to detect confounders, e.g., collect accompanying patient data from referring hospitals/physicians and details on the (quality of the) treatment given. Viewed differently, the purpose of data collection, regardless whether it is data from randomized clinical trials or routine care, is to improve treatment quality for all patients. Peters et al. (2010)[20] show that even within clinical trials treatment quality is highly variable among institutes, i.e. institutes treating fewer patients delivering lower quality treatments. Given these difference, it is debatable whether conclusions drawn from trials which were conducted at selected institutes translate into routine clinical care where the standard of care may be generally lower and patient populations differ[21,22]. Data collected in routine clinical care is directly sampled from the population in question unlike trial data derived from a biased proxy. Therefore, systematic data collection in routine clinical care will not only provide new opportunities for further analyses (with the abovementioned necessary caution) but it will also allow systematic studies of the general patient population and tracking whether treatment benefits observed in clinical trials arrived in routine clinical care.

Continued concerns over patient privacy might render institutes reluctant to participate in systematic data sharing. Illegal access to data is prevented within the euroCAT learning environment: the web browser-based learning interface is only accessible with registered user accounts and learning runs are always linked to such account. Learning algorithms circulating in the network need to be authenticated by a digital file signer that is available only to registered members of the euroCAT network. Furthermore, permission to learn on an institute's data is granted by the respective institute's principal investigators per user account or on a run-by-run basis. Additionally, illegal data transfers can be identified and shut down: standard master/site communication is limited to small volumes like model parameters, prediction outcomes, and summary statistics. Limits on the communication volume therefore render high volume data transfers impossible. Collaboration with external parties always comes with a risk of losing control over one's data. Mutual trust and legal assurance to safeguard other parties' data are key aspects in scientific collaboration. However, in comparison to the traditional

data exchange collaborations, a data sharing network such as euroCAT adds technical control mechanisms to manage and limit access to an institute's data.

The pilot study was restricted to sharing a dataset of limited size in three countries. However, the range of variables, number of patients, and number of institutes is variable: linking an entire hospital's EHR and PACS to the learning environment is theoretically possible. Further, the ontologies used for euroCAT to match variables across institutes bear the potential to facilitate data sharing around the globe. For euroCAT, data was shared across clinics located in three different countries, i.e. with three different national data collection guidelines and three different languages (Dutch, French, and German). This pilot study has led to followup projects in, among others, the Netherlands (duCAT), Italy (VATE), the USA (meerCAT), Australia (ozCAT), Canada (canCAT), and China (sinoCAT).

The potential of the euroCAT infrastructure exceeds the presented results. The capability to learn SVMs is just one example for applications of the distributed learning infrastructure. The ADMM algorithm used for SVMs is extendable to other existing machine learning methods like linear/logistic regressions and feature selection methods like (logistic) LASSO[10]. Independent of the ADMM algorithm, the infrastructure can facilitate other machine learning techniques such as Bayesian Networks learned from distributed data[23]. More generally, any desirable computation requiring access to an institute's data with subsequent aggregation on the master is feasible. Systematic data sharing efforts such as euroCAT will likely profit from the ongoing research in the flourishing fields of machine learning and artificial intelligence.
The presented IT infrastructure facilitates modeling of multicentric data without direct access to said data. This method bears the risk that inter-institutional bias in variables, e.g., due to inconsistent (toxicity) scoring, varying reporting standards, different patient populations, or data collection errors remain unnoticed. Future work will be focused on the systematic detection of such affected data in a distributed learning network.

## Conclusion

Multi-centric rapid learning for health care is feasible as shown by the support vector machines developed in the euroCAT network. We have no doubts that the clinical decision support systems of the future would routinely use models based on data available in distributed databases *across* national borders. One solution for surmounting accompanying technical, legal, and ethical issues with data sharing is already delivered across three countries by the euroCAT system and has shown to scale globally. We believe that distributed learning is the best way to go for building clinically reliable models that are universally applicable, personalized, and robust.

## Funding

Authors acknowledge financial support from ERC advanced grant (ERC-ADG-2015, n° 694812 – Hypoximmuno). This research is also supported by the Dutch technology Foundation STW (grant n° 10696 DuCAT & n° P14-19 Radiomics STRaTegy), which is the applied science division of NWO, and the Technology Programme of the Ministry of Economic Affairs. Authors also acknowledge financial support from the EU 7th framework program (ARTFORCE – n° 257144, REQUITE – n° 601826), SME Phase 2 (EU proposal 673780 – RAIL), EUROSTARS (SeDI, CloudAtlas, DART), the European Program H2020-2015-17 (BD2Decide – PHC30-689715 and ImmunoSABR – n° 733008), Kankeronderzoekfonds Limburg from the Health Foundation Limburg, Alpe d'HuZes-KWF (DESIGN), and the Dutch Cancer Society. This publication was supported by the Dutch national program COMMIT (Prana Data project).

## Declaration of interests

Timo Deist holds a part-time employment with ptTheragnostic BV, The Netherlands. Philippe Lambin is in the advisory board of ptTheragnostic BV, The Netherlands.

## Acknowledgement

We would like to thank Varian Medical Systems for providing the distributed learning manager, and Wolfgang Wiessler and Tim Hendriks for their dedicated support.

## References


1. Lambin P *et al.* Predicting outcomes in radiation oncology–multifactorial decision support systems. *Nat Rev Clin Oncol* 2013;**10**(1):27–40.
2. Mackin D *et al.* Measuring computed tomography scanner variability of radiomics features. *Invest Radiol* 2015;**50**(11):757–65.
3. Rosewall T *et al.* Inter-professional variability in the assignment and recording of acute toxicity grade using the RTOG system during prostate radiotherapy. *Radiother Oncol* 2009;**90**(3):395–9.
4. Dekker A *et al.* Rapid learning in practice: a lung cancer survival decision support system in routine patient care data. *Radiother Oncol* 2014;**113**(1):47–53.
5. Lambin P *et al.* 'Rapid learning health care in oncology' – an approach towards decision support systems enabling customised radiotherapy'. *Radiother Oncol* 2013;**109**(1):159–64.
6. Constable SD, Tang Y, Wang S, Jiang X, Chapin S. Privacy-preserving GWAS analysis on federated genomic datasets. *BMC Med Inform Decis Mak* 2015;**15**(5):S2.
7. Jiang W *et al.* WebGLORE: a web service for Grid LOgistic REgression. *Bioinformatics* 2013:btt559.
8. Welcome to the NCBO BioPortal | NCBO BioPortal. [Online] Available: <http://bioportal.bioontology.org/>; 2016 [accessed 26.10.16].
9. Prud'Hommeaux E, Seaborne A. SPARQL query language for RDF. *W3C Recomm.*, vol. **15**; 2008.
10. Boyd S, Parikh N, Chu E, Peleato B, Eckstein J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found Trends Mach Learn* 2011;**3**(1):1–122.
11. Distributed optimization and statistical learning via the alternating direction method of multipliers. [Online] Available: <http://web.stanford.edu/~boyd/papers/admm_distr_stats.html>; 2016 [accessed: 26.10.16].
12. Dehing-Oberije C, Ruysscher DD, van Baardwijk A, Yu S, Rao B, Lambin P. The importance of patient characteristics for the prediction of radiation-induced lung toxicity. *Radiother Oncol* 2009;**91**(3):421–6.
13. Nalbantov G *et al.* Cardiac comorbidity is an independent risk factor for radiation-induced lung toxicity in lung cancer patients. *Radiother Oncol J Eur Soc Ther Radiol Oncol* 2013;**109**(1):100–6.
14. Prechelt L. Early stopping — but when? In: Montavon G, Orr GB, Müller K-R, editors. *Neural networks: tricks of the trade.* Berlin Heidelberg: Springer; 2012. p.53–67.
15. Oberije Cary *et al.* EuroCAT umbrella protocol for NSCLC, 2013.
16. Meldolesi E *et al.* An umbrella protocol for standardized data collection (SDC) in rectal cancer: a prospective uniform naming and procedure convention to support personalized medicine. *Radiother Oncol J Eur Soc Ther Radiol Oncol* 2014;**112**(1):59–62.
17. Abernethy AP *et al.* Rapid-learning system for cancer care. *J Clin Oncol Off J Am Soc Clin Oncol* 2010;**28**(27):4268–74.
18. McGale P, Cutter D, Darby SC, Henson KE, Jagsi R, Taylor CW. Can observational data replace randomized trials? *J Clin Oncol* 2016;**34**(27):3355–7.
19. Chavez-MacGregor M, Giordano SH. Randomized clinical trials and observational studies: is there a battle? *J Clin Oncol* 2016;**34**(8):772–3.
20. Peters LJ *et al.* Critical impact of radiotherapy protocol compliance and quality in the treatment of advanced head and neck cancer: results from TROG 02.02. *J Clin Oncol Off J Am Soc Clin Oncol* 2010;**28**(18):2996–3001.
21. Murthy VH, Krumholz HM, Gross CP. Participation in cancer clinical trials: race-, sex-, and age-based disparities. *JAMA* 2004;**291**(22):2720–6.
22. Movsas B *et al.* Who enrolls onto clinical oncology trials? a radiation patterns of care study analysis. *Int J Radiat Oncol Biol Phys* 2007;**68**(4):1145–50.
23. Jochems A *et al.* Distributed learning: developing a predictive model based on data from multiple hospitals without data leaving the hospital – a real life proof of concept. *Radiother Oncol* 2016.
24. Platt JC. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *In: Advances in large margin classifiers.* p.61–74.

## Appendix A

A support vector machine determines two parallel hyperplanes, forming a 'border' which separates the feature space into two large regions and a margin between the planes. Each dimension of this feature space represents one patient feature (e.g., FEV1 (in %) or cardiac comorbidity) and each patient is represented by one point in this space. For simple problems, the intention would be to identify hyperplanes that separate all patients with dyspnea from the group of patients without dyspnea. This is in not possible in most cases, therefore the objective becomes to find hyperplanes such that

- most of the dyspneic patients are on one and non-dyspneic patients are on the other side; if there is a patient on the 'wrong' side of the border, the distance to the border is as small as possible;
- the border between the groups of dyspneic and non-dyspneic patients is as large as possible.

The optimal hyperplanes are determined by

$$\min_{w,b} \frac{1}{\lambda} ||w||_2^2 + \sum_{i=1}^{n} s_i \tag{1}$$

$$\text{such that } y_i(d_i w + b) \geq 1 - s_i \text{ for all } i = 1, \dots, n \tag{2}$$

$$s_i \geq 0 \text{ for all } i = 1, \dots, n. \tag{3}$$

$w$ is the normal vector of the separating hyperplanes, $b$ is the bias term. $(w,b)$ characterizes the hyperplanes. $s_i$ is an auxiliary variable for sample $i$ representing the classification error. $\lambda$ is a parameter to assign more importance to the first or the second term of the objective. $\lambda$ needs to be positive. $y_i \in \{-1,1\}$ is the label of training sample $i$. $d_i$ is the vector of features for sample $i$. Minimizing the first term in the objective function, $\frac{1}{\lambda}||w||_2^2$, maximizes in the margin, i.e., the space between both hyperplanes. Minimizing $\sum_{i=1}^{n} s_i$ minimizes the classification error. The objective is split into two terms, $\frac{1}{\lambda}||w||_2^2$ and $\sum_{i=1}^{n} s_i$: The latter is separable among data samples such that the value for $\sum_{i=1}^{n} s_i$ can be obtained by slicing up the dataset into multiple parts, computing the contribution of each slice independently and merging the results afterwards. This property (and other) can be exploited such that the SVM optimization problem is solvable in a distributed fashion. Boyd et al.(2011)[10] discuss a distributed formulation of a support vector machine using the Alternating Direction Method of Multipliers and provide MATLAB code[11]. The ADMM algorithm gained popularity in the machine learning community as it allows to split up large datasets into smaller portions and distribute the analysis over multiple machines. In our multi-centric learning context, the same property is exploited to overcome the restriction that data may not be centralized. The formulation is

$$x_j^{k+1} = \text{argmin}_{x_i}\left(1^T(A_i x_i + 1)_+ + \left(\frac{\rho}{2}\right)\|x_i - z^k + u_i^k\|_2^2\right) \quad (4)$$

$$\hat{x}_j^{k+1} = \alpha x_j^{k+1} + (1-\alpha)z^k \quad (5)$$

$$z^{k+1} = \frac{\rho}{\left(\frac{1}{\lambda}\right) + N\rho}\left(\bar{\hat{x}}^{k+1} + \bar{u}^k\right) \quad (6)$$

$$u_j^{k+1} = u_j^k + \hat{x}_j^{k+1} - z^{k+1} \quad (7)$$



where $x = (w,b)$, $N$ is the number of sites, and $p$ and $\alpha$ are model parameters. In each iteration $k + 1$, $x_j^{k+1}$ is computed at each site $j$ and transmitted to the master. At the master, a relaxation function (5) is applied to $x_j^{k+1}$ yielding $\hat{x}_j^{k+1}$.The average $\bar{\hat{x}}^{k+1}$ of all sites is used to compute $z^{k+1}$ and $u^{k+1}$, which are transmitted to the sites and are used as input for the computation of $x_j^{k+2}$ in the next iteration. $x_j^{k+1}$ is calculated to reduce the classification error, $z^{k+1}$ is calculated to increase the margin, and $u_j^{k+1}$ is the dual variable inherent to the ADMM algorithm. ADMM requires a multitude of iterations in which estimates for $(w,b)$ are refined using each site's data. Once an estimate of $(w,b)$ is chosen and the algorithm is stopped, Platt scaling[24] is applied: the values $d_i\ w+b$ per training sample i are fitted to the dyspnea outcomes using a logistic regression. $d_i\ w+b$ is a measure of training sample $i$'s location in space relative to the two hyperplanes. The logistic regression equation allows to assign a dyspnea probability to patients in the training and validation datasets.

## Appendix B

Data processing was done in MATLAB (MathWorks, Natick, MA, USA). Pseudocodes of the MATLAB functions executed on the master and sites are shown in Figs. B1 and B2, respectively. Patient features were rescaled before learning to improve algorithm performance. A variable $v$ was rescaled to $\tilde{v}$ according to

$$\tilde{v} = \frac{v - \min(v)}{\max(v) - \min(v)}$$

where min $(v)$ and max $(v)$ are minimal and maximal feature values, respectively, found within the entire learning network. This step requires centralizing minimal and maximal feature values for each site. This poses no threat to patient privacy since no value can be allocated to a single patient assuming that each site's database contains more than one patient. Future work should be dedicated to replacing this normalization by a generally privacy-preserving method.

The categorical variables cardiac comorbidity and chemotherapy timing were each coded as ($c$-1) dummy variables, $c$ being equal to the variable's cardinality.

Missing values were imputed using the mean for continuous variables and mode for categorical variables. Means and modes were derived per site.

The code designed to guide the machine learning process within the IT infrastructure is available on **www.eurocat.info** with further information about the infrastructure and how to join the CAT project.

The chosen model parameters are ρ=1, α=1.5, and λ=0.01. The convergence criteria are set as described by[11] with absolute tolerance = $10^{-4}$ and relative tolerance = $10^{-2}$. $x$, $z$, and $u$ are initialized at the zero vector. Parameters have been set manually and based on choices found in[11]. Future work on deriving clinically-relevant prediction models exceeding an exemplary nature should also comprise systematic parameter tuning.

```
read user input file
IF in first iteration
    assign master and sites to 'data reading' stage
    create input files for sites
ELSE
    read site output files
    IF in 'data reading' stage
        assign master and sites to 'learning' stage
        compute min. and max. values per variable over all sites
        assign min. and max. values as input for sites
        create input files for sites
    ELSEIF in 'learning' stage
        compute z- and u-updates
        check convergence criteria
        IF optimization has converged OR iteration limit is reached
            assign master and sites to 'evaluation' stage
            set final model as mean of x over all sites
            assign x̄ as input for sites
        END
        create input files for sites
    ELSEIF in 'evaluation' stage
        fit logistic regression to d_i w + b and y_i for all training site samples i
        compute regression estimates for all training and validation site samples
        write regression estimates and y to result file
    END
END
```

**Figure B.1.** Pseudocode of the MATLAB function executed on the master.

```
read master output file
IF in 'data reading' stage
    read data from site
    compute min. and max. values per variable
    assign min. and max. values as input for master
ELSEIF in 'learning' stage
    IF processed data file exists
        read processed data file
    ELSE
        read data from site
        impute missing data
        dummy code categorical data
        rescale data using min. and max. values provided by master
        write processed data to file
    END
    IF this is a training site
        compute x-update
    END
ELSEIF in 'evaluation' stage
    read processed data file
    compute d_i w + b for each sample i
    assign the pairs (d_i w + b, y_i) for each sample i as input for master
END
create input file for master
```

**Figure B.2.** Pseudocode of the MATLAB function executed on the sites.

# Chapter 3

## Distributed learning on 20 000+ lung cancer patients – The Personal Health Train

Timo M. Deist, Frank J.W.M. Dankers, Priyanka Ojha, M. Scott Marshall, Tomas Janssen, Corinne Faivre-Finn, Carlotta Masciocchi, Vincenzo Valentini, Jiazhou Wang, Jiayan Chen, Zhen Zhang, Emiliano Spezi, Mick Button, Joost Jan Nuyttens, René Vernhout, Johan van Soest, Arthur Jochems, René Monshouwer, Johan Bussink, Gareth Price, Philippe Lambin, Andre Dekker

*Submitted*

**Abstract**

Access to healthcare data is indispensable for scientific progress and technological innovation. Sharing healthcare data is time-consuming and notoriously difficult due to privacy and regulatory concerns. The Personal Health Train provides a privacy-by-design infrastructure connecting FAIR (Findable, Accessible, Interoperable, Reusable) data sources and allows distributed data analysis and machine learning. Patient-specific data never leaves a healthcare provider. We present results of the Personal Health Train's application across eight international healthcare institutes (Amsterdam, Cardiff, Maastricht, Manchester, Nijmegen, Rome, Rotterdam, Shanghai): a study executed and analyzed in only 4 months. We trained and validated a distributed logistic regression model predicting post-treatment two-year survival in 23 203 non-small cell lung cancer patients treated between 1978-2015 based on tumor staging definitions established by the American Joint Committee on Cancer. The Personal Health Train infrastructure demonstrably overcomes patient-privacy barriers to healthcare data sharing and promotes global evidence-based medicine.

Many current innovations in medicine, including personalized medicine, artificial intelligence, (big) data-driven medicine, learning healthcare systems, value-based healthcare and decision support systems, rely on the sharing of data across healthcare providers. Conventional data analysis requires sharing and centralization of data to answer research questions. However, data sharing is hampered by administrative, political, ethical, and technical barriers[1]. This limits the amount of healthcare data available for life sciences in general as well as for other secondary uses such as healthcare quality assurance.



Distributed (machine) learning reformulates conventional data analysis algorithms so that data centralization becomes unnecessary. Distributed algorithms iteratively analyze separate databases and return the same solution as if data were centralized: essentially sharing research questions and answers between databases instead of data.

We are convinced that only sharing research questions (and answers) between healthcare providers is a better, sustainable approach to medical data analysis, and can unlock orders of magnitude more data without violating privacy. To this end, we have developed an infrastructure called the Personal Health Train[2] (PHT) consisting of

- sites ("stations") containing FAIR[3] (Findable, Accessible, Interoperable, Reusable) data,
- technical network connections and legal frameworks ("tracks"),
- statistical learning applications ("trains").

A global community of likeminded healthcare providers and academic partners called CORAL (Community in Oncology for RApid Learning) was initiated at the 2016 European Society for Radiotherapy and Oncology (ESTRO) conference. In various research projects across the globe, CORAL members have worked on the realization of the PHT.

An infrastructure to bring research questions to the data has been demonstrated to work recently in projects such as euroCAT[4,5], DataSHIELD[6] and OHDSI[7]. However, challenges remain in terms of the number of data subjects, number of data providers, and global coverage.

The aim of this study is to show that the PHT distributed learning infrastructure can be scaled to many thousands of patients, approaching the size of national healthcare registries. Specifically, we set the goal (as registered on clinicaltrials.gov[8]) to machine learn a predictive model for post-treatment two-year survival on more than 20 000 non-small cell lung cancer (NSCLC) patients, in at least five healthcare providers from more than five countries—without any patient data leaving a healthcare provider.

## Results

In total, eight healthcare providers ("stations") were contacted on 18-06-2018 and two additional sites were contacted later. At the deadline of 01-09-2018 (71 days after the first formal project invitation), eight sites (in Amsterdam, Cardiff, Maastricht, Manchester, Nijmegen, Rome, Rotterdam, Shanghai) made NSCLC patient data available in their local database endpoints and two sites did not participate for logistical reasons: delayed response to first formal invitation in one case and too little time to participate after a second round of invitations in another case. NSCLC patient data consists of two-year survival information

- diagnosis,
- diagnosis date,
- survival follow-up status,
- survival follow-up status date,

and cancer staging as defined by the American Joint Committee on Cancer (AJCC, see Methods for details)

- tumor (T) stage,
- lymph node (N) stage,
- metastasis (M) stage,
- overall disease stage.

### *Data availability*

A summary statistics application ("train") was sent via the Varian Learning Portal ("track"). It computed patient counts for each variable category, displayed in Table 1. Each site confirmed the validity of the summary statistics, a quality control step to ensure that correct data was used for modelling. A total number of 37 090 patients became available in the system. When restricting the search to patients:

- diagnosed or treated from 01-01-1978 (effective date of the AJCC TNM cancer staging edition 1) and before 01-01-2016 (allowing at least two years survival follow-up),
- with complete diagnosis date, follow-up date, and follow-up status (to calculate two-year survival),

the number of available patients decreased to 28 178, which forms the *modelling cohort*. Data of patients diagnosed before 2005 were mainly collected by two sites (with minor contributions from two other sites). Data of patients diagnosed after 2005 were made available by all sites. Overall, recent data was more abundant. More than half of the modelling data was provided by two sites: site G (43.0%) and site E (17.0%). Less than 6% of the modelling data was sourced from three sites: site D (2.4%), site C (2.3%), and site B (1.0%).

*Modelling cohort distribution*

Histograms for T, N, M, and overall stage categories after binning into supercategories (Table 6) but before imputation are shown in Figure 1. Patients with missing or right-censored two-year survival are excluded. The histograms are separate per site (x-axis) and split for patients alive and dead at two years after diagnosis (above and below x-axis). Patient counts are normalized per site. Sites are ordered by the percentage of patients alive at two years.

The percentage of patients alive at two years differed greatly in the provided data across sites (Figure 1): from 89.1% in site A to 18.8% in site H. The distribution of T, N, M, and overall stage categories also varied across sites. Notably, T1 clearly dominated in sites A and C but other sites display a more balanced distribution of T categories (Figure 1a). In sites A-E, N0 is the modal lymph node category but N2 is most frequent in sites F-H (Figure 1b). All sites report most patients in the M0 category but the decrease in M0 patients correlates loosely with the percentage of patients alive at two years per site, e.g., site H reports 41.4% M1 compared to 8.8% in site A (Figure 1c). As a direct consequence of the differences in T, N, and M category distributions, the overall stage distribution varies across sites (Figure 1d).

In general, data completeness is not consistent in the network (Table 1). Sufficient follow-up information to compute two-year survival ranges from 92.1% (site D) to 44.1% (site B). Note that patients with incomplete follow-up (right-censored) have not been included in the modelling cohort displayed in Figure 1 and Figure 2. T, N, M, or overall stage information is frequently missing in half of the sites (sites E-H). Overall stage categories are not always reported: sites E and H do not provide overall stage information. Sites G, F, and A miss it for 39.8%, 31.8%, and 2.2% of their patients, respectively.

*Distributed machine learning*

Based on the temporal distribution of patients in the modelling cohort, we selected patients from 01-01-1978 until and including 31-12-2011 for training and patients from 01-01-2012 until and including 31-12-2015 for validation so that we achieved a split of approximately 2/3 to 1/3. We selected a temporal split for training and validation (TRIPOD type 2b validation[9]) to simulate the development of the model on historical patient data and subsequent application in future patients.

Only 14 660 patients of 28 178 patients were complete cases (T, N, M, overall stage, and two-year survival) in the modelling cohort (Table 3). Missing T, N, M and overall stage were imputed using logical rules according to the AJCC TNM cancer staging editions and observed patient frequencies in the respective site. Imputation did not result in complete cases for a subset of patients (see methods section for details) yielding 14 810 (63.8%) patients for training and 8 393 (36.2%) patients for validation, a total of 23 203 patients.

The logistic regression application trained a model from the training data (years 1978-2011) with coefficients as displayed in Table 2. The convergence criteria of the algorithm are met after 81 iterations (25 minutes). The convergence of the algorithm is displayed in Figure 2b: the root mean square error (RMSE) for predicting the probability of two-year survival (left y-axis) in the training cohort decreases per iteration and approaches 0.42. Although the RMSE has stabilized, not all regression coefficients (right y-axis) have converged.

The validation application assessed the model's performance on the validation cohort (years 2012-2015). The validation performance is described by the combined RMSE for patients from all sites (Figure 2b), the receiver operating characteristic (ROC) curve per site and their corresponding areas under the curve (AUCs) (Figure 2c), and by an exemplary

calibration plot of the site with most patient data provided for training and validation (site G, Figure 2d). Calibration plots for all other sites are displayed in Figure S1 (Supplementary Information). Table 3 summarizes patient counts (available in the system and in the modelling cohort before and after imputation) and model performance per site. The validation RMSE almost-monotonically decreases during optimization on the training cohort. Discriminative performance of the model (as measured by the AUC), varies across sites from 0.85 (site A) to 0.58 (site D). Model calibration in site G is good with a calibration-in-the-large of 0.02 and calibration-slope of 0.75 but calibration varies strongly across sites. For example, site A (Supplementary Information, Figure S1) displays a calibration-in-the-large of 2.39 and a calibration slope of 1.09.

**Table 1.** Summary statistics of all patients provided by the sites. These are patient counts before filtering for the modelling cohort (diagnosed in 1978-2015 with available two-year survival data and at least one stage variable) and before imputation.

| | Site A | Site B | Site C | Site D | Site E | Site F | Site G | Site H | | Site A | Site B | Site C | Site D | Site E | Site F | Site G | Site H |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Disease** | | | | | | | | | **Overall stage** | | | | | | | | |
| NSCLC | 5214 | 706 | 829 | 785 | 6211 | 4110 | 16260 | 2975 | Missing | 92 | 3 | 0 | 0 | 6211 | 1714 | 7573 | 2975 |
| **T stage** | | | | | | | | | 0 | 208 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| Missing | 4 | 20 | 0 | 0 | 77 | 807 | 6703 | 10 | I | 0 | 0 | 0 | 0 | 0 | 152 | 282 | 0 |
| T0 | 6 | 1 | 0 | 2 | 3 | 36 | 1 | 16 | IA | 2413 | 93 | 0 | 141 | 0 | 31 | 704 | 0 |
| T1 | 650 | 30 | 34 | 74 | 322 | 429 | 674 | 200 | IA1 | 0 | 0 | 35 | 0 | 0 | 0 | 6 | 0 |
| T1a | 1694 | 82 | 35 | 42 | 337 | 56 | 351 | 78 | IA2 | 0 | 0 | 191 | 0 | 0 | 0 | 36 | 0 |
| T1b | 588 | 40 | 191 | 88 | 285 | 96 | 313 | 117 | IA3 | 0 | 0 | 185 | 0 | 0 | 0 | 31 | 0 |
| T1c | 0 | 1 | 185 | 0 | 15 | 16 | 73 | 16 | IB | 501 | 48 | 104 | 141 | 0 | 56 | 373 | 0 |
| T2 | 110 | 75 | 39 | 128 | 1079 | 803 | 2138 | 844 | II | 0 | 0 | 0 | 0 | 0 | 75 | 101 | 0 |
| T2a | 1032 | 92 | 104 | 139 | 772 | 132 | 472 | 91 | IIA | 459 | 13 | 49 | 65 | 0 | 17 | 135 | 0 |
| T2b | 206 | 18 | 49 | 50 | 194 | 65 | 227 | 45 | IIB | 188 | 56 | 39 | 78 | 0 | 56 | 235 | 0 |
| T3 | 303 | 165 | 77 | 109 | 1460 | 523 | 1936 | 518 | III | 0 | 0 | 0 | 2 | 0 | 52 | 621 | 0 |
| T4 | 254 | 151 | 107 | 143 | 1667 | 1037 | 1932 | 639 | IIIA | 786 | 187 | 110 | 215 | 0 | 348 | 1689 | 0 |
| TX | 164 | 31 | 8 | 10 | 0 | 108 | 1439 | 396 | IIIB | 104 | 103 | 116 | 103 | 0 | 577 | 1753 | 0 |
| Tis | 203 | 0 | 0 | 0 | 0 | 2 | 1 | 5 | IIIC | 0 | 0 | 0 | 1 | 0 | 1 | 18 | 0 |
| **N stage** | | | | | | | | | IV | 199 | 198 | 0 | 39 | 0 | 1012 | 2553 | 0 |
| Missing | 0 | 20 | 0 | 0 | 14 | 821 | 6705 | 7 | IVA | 75 | 0 | 0 | 0 | 0 | 4 | 54 | 0 |
| N0 | 3649 | 255 | 637 | 384 | 2756 | 1041 | 2830 | 660 | IVB | 189 | 5 | 0 | 0 | 0 | 15 | 95 | 0 |
| N1 | 520 | 49 | 13 | 153 | 635 | 208 | 598 | 180 | **Diagnosis year** | | | | | | | | |
| N2 | 777 | 271 | 143 | 215 | 1835 | 1132 | 3510 | 977 | 1950-1959 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| N3 | 141 | 83 | 36 | 23 | 971 | 810 | 1437 | 600 | 1960-1969 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| NX | 127 | 28 | 0 | 10 | 0 | 98 | 1180 | 551 | 1970-1979 | 0 | 0 | 0 | 0 | 0 | 0 | 693 | 1 |
| **M stage** | | | | | | | | | 1980-1989 | 0 | 0 | 0 | 0 | 0 | 3 | 2301 | 362 |
| Missing | 2 | 3 | 0 | 0 | 0 | 554 | 6705 | 4 | 1990-1999 | 0 | 2 | 0 | 0 | 1 | 16 | 3192 | 809 |
| M0 | 4742 | 491 | 829 | 734 | 4799 | 2073 | 6435 | 1526 | 2000-2004 | 0 | 5 | 0 | 8 | 1 | 74 | 1527 | 421 |
| M1 | 87 | 70 | 0 | 8 | 650 | 1253 | 1926 | 1053 | 2005 | 0 | 18 | 12 | 51 | 223 | 185 | 374 | 83 |
| M1a | 92 | 7 | 0 | 11 | 246 | 36 | 164 | 19 | 2006 | 0 | 15 | 31 | 50 | 313 | 248 | 365 | 78 |
| M1b | 285 | 121 | 0 | 20 | 510 | 124 | 497 | 107 | 2007 | 1 | 24 | 44 | 59 | 276 | 275 | 506 | 68 |
| M1c | 1 | 5 | 0 | 0 | 6 | 15 | 107 | 29 | 2008 | 190 | 123 | 48 | 51 | 314 | 282 | 498 | 95 |
| MX | 5 | 9 | 0 | 12 | 0 | 55 | 426 | 237 | 2009 | 214 | 127 | 71 | 42 | 348 | 317 | 528 | 99 |
| **2-year survival** | | | | | | | | | 2010 | 318 | 92 | 100 | 62 | 401 | 338 | 541 | 125 |
| Missing/ Right-censored | 614 | 395 | 164 | 62 | 692 | 818 | 3412 | 477 | 2011 | 445 | 33 | 117 | 77 | 455 | 306 | 554 | 120 |
| No | 464 | 112 | 258 | 396 | 3834 | 2305 | 9357 | 2048 | 2012 | 557 | 32 | 112 | 78 | 626 | 300 | 603 | 121 |
| Yes | 4136 | 199 | 407 | 327 | 1685 | 987 | 3491 | 450 | 2013 | 690 | 34 | 97 | 75 | 692 | 369 | 697 | 100 |
| | | | | | | | | | 2014 | 971 | 52 | 31 | 70 | 573 | 345 | 755 | 110 |
| | | | | | | | | | 2015 | 1057 | 43 | 62 | 63 | 641 | 300 | 763 | 112 |
| | | | | | | | | | 2016 | 761 | 37 | 103 | 58 | 666 | 302 | 744 | 136 |
| | | | | | | | | | 2017 | 0 | 35 | 1 | 41 | 562 | 308 | 607 | 112 |
| | | | | | | | | | 2018 | 10 | 11 | 0 | 0 | 118 | 142 | 163 | 23 |

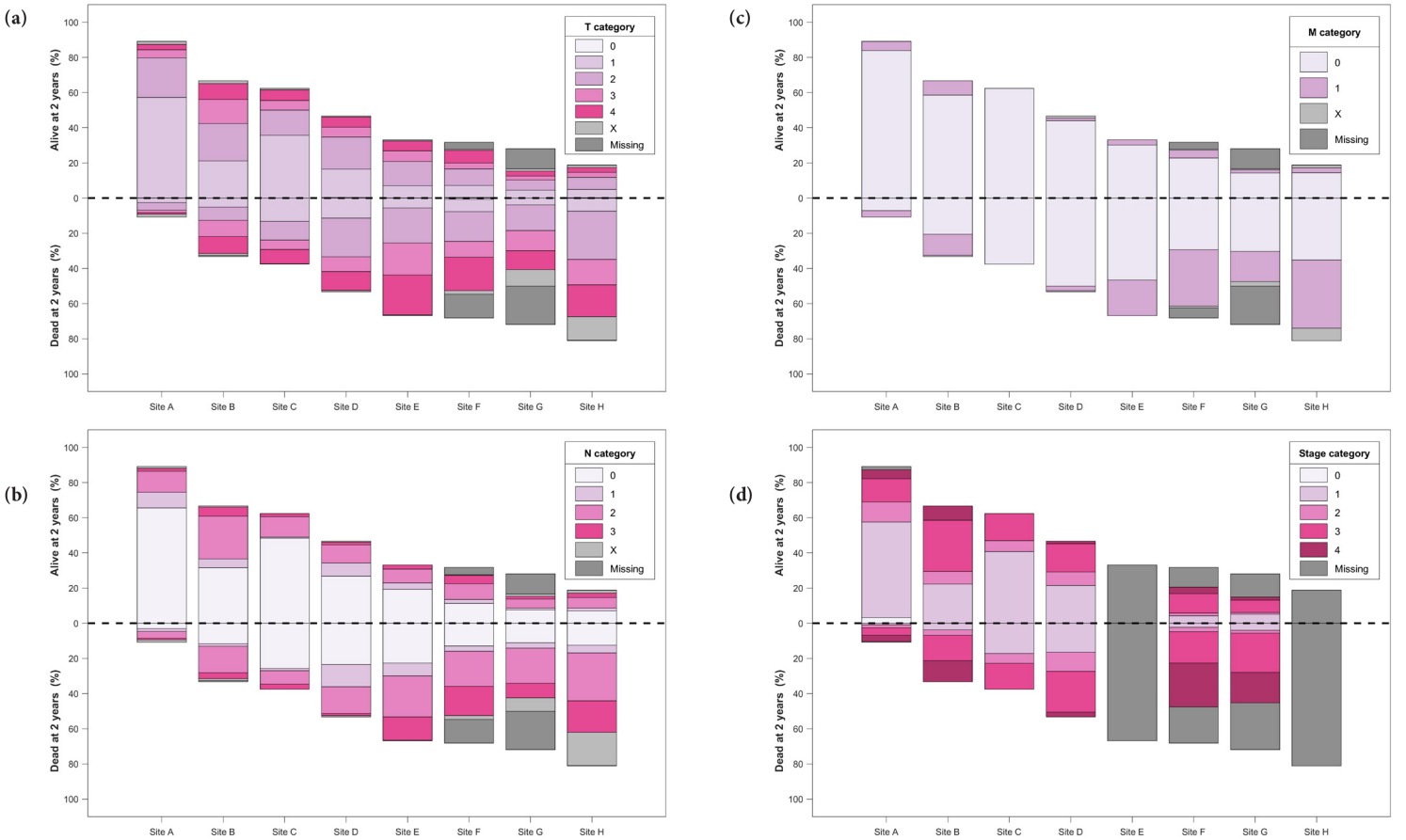



**Figure 1.** Distributions of T, N, M, and overall stage supercategories (**a**, **b**, **c**, **d**, respectively) for patients available for training or validation per site (i.e. the modelling cohort) before selecting for complete cases and imputation. Patients with missing or right-censored two-year survival are excluded. The histograms are separate per site (x-axis) and split for patients alive and dead at two years after diagnosis (above and below x-axis). Patient counts are normalized per site. The vertical position of the entire bar indicates the two-year survival ratio of each site.

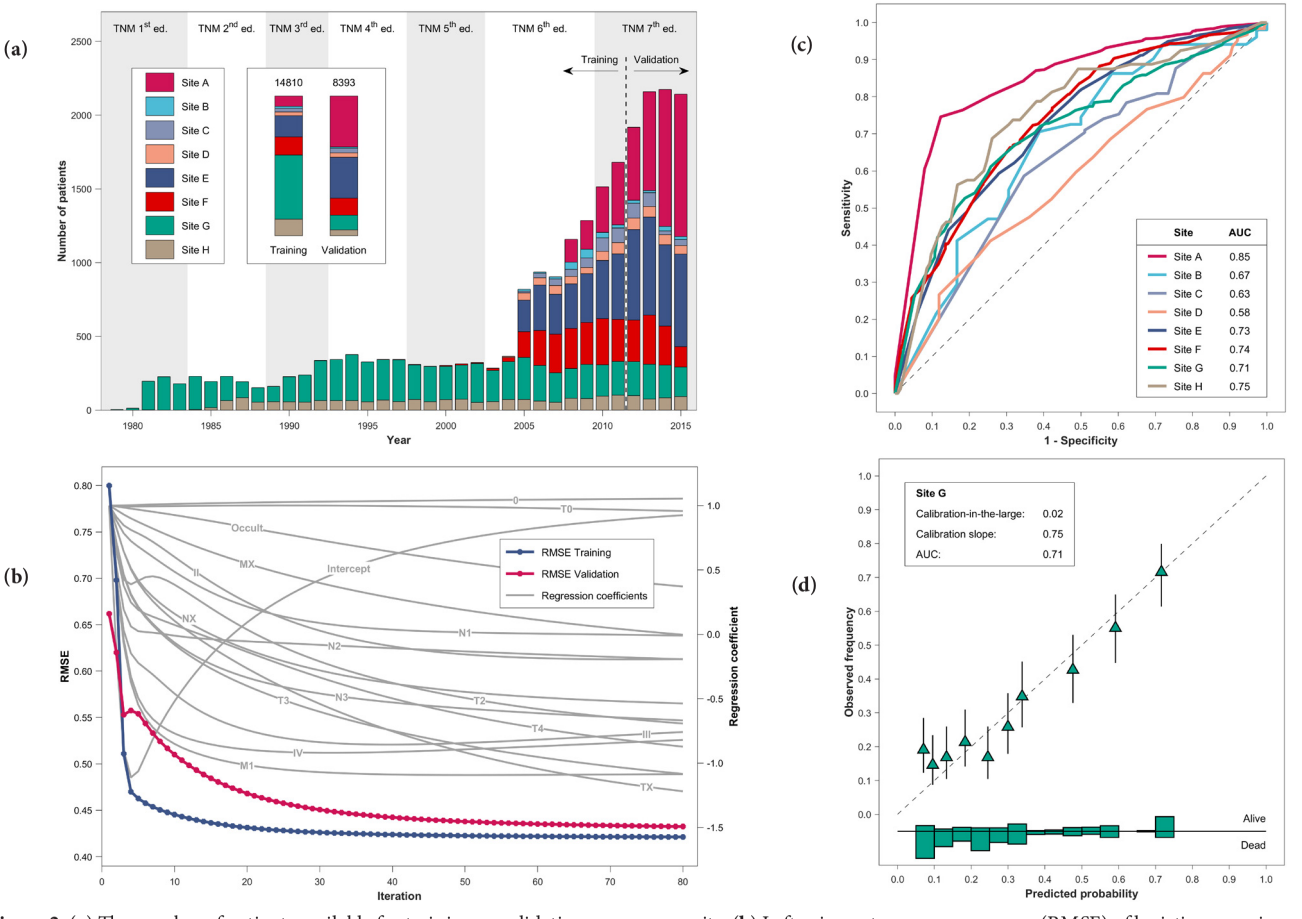



**Figure 2.** (**a**) The number of patients available for training or validation per year per site. (**b**) Left axis: root mean square error (RMSE) of logistic regression models optimized on the training cohort at a given iteration for training and validation cohorts. Right axis: regression coefficients for T, N, M, and overall stage categories computed by the ADMM algorithm at a given iteration. (**c**) Receiver operating characteristic curves with area under the curve (AUC) values for the validation cohort per site. (**d**) Calibration plot of the validation cohort for site G, the site with most training and validation data. Calibration plots for the remaining sites are displayed in Figure S1.

**Table 2.** Logistic regression coefficients per supercategory. T1N0M0 and overall stage category I is the reference case.

| Intercept | | T | | N | | M | | Overall stage | |
|---|---|---|---|---|---|---|---|---|---|
| 0.93 | **0** | 0.96 | **0** | ref. | **0** | ref. | **0** | 1.05 |
| | **1** | ref. | **1** | -0.01 | **1** | -1.09 | **I** | ref. |
| | **2** | -0.69 | **2** | -0.19 | **X** | 0.00 | **II** | -0.19 |
| | **3** | -1.08 | **3** | -0.67 | | | **III** | -0.76 |
| | **4** | -0.87 | **X** | -0.54 | | | **IV** | -0.82 |
| | **X** | -1.22 | | | | | **Occult** | 0.37 |

**Table 3.** Patient counts and model performance per site. Sites E and H are listed as incomplete as neither site published overall staging data (which may be imputed from T, N and M stages). AUC: area under the receiver operating characteristic curve. CI: confidence interval.

| Site | Available patients | Modelling cohort patient counts (complete cases, 1978-2015) | | | | | | Model performance | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Before imputation | | | After imputation | | | Training | | Validation | | | |
| | | Training | Validation | Total | Training | Validation | Total | AUC | 95%-CI | AUC | 95%-CI | Calibration-in-the-large | Calibration slope |
| Site A | 5214 | 1050 | 3024 | 4074 | 1084 | 3058 | 4142 | 0.79 | [0.75, 0.82] | 0.85 | [0.83, 0.87] | 2.39 | 1.09 |
| Site B | 706 | 203 | 87 | 290 | 204 | 87 | 291 | 0.71 | [0.62, 0.77] | 0.67 | [0.54, 0.78] | 1.04 | 0.62 |
| Site C | 829 | 390 | 260 | 650 | 390 | 260 | 650 | 0.62 | [0.57, 0.67] | 0.63 | [0.57, 0.69] | 0.36 | 0.59 |
| Site D | 785 | 398 | 276 | 674 | 398 | 276 | 674 | 0.61 | [0.55, 0.66] | 0.58 | [0.51, 0.64] | 0.07 | 0.40 |
| Site E | 6211 | 0 | 0 | 0 | 2265 | 2458 | 4723 | 0.70 | [0.68, 0.72] | 0.73 | [0.70, 0.75] | -0.09 | 0.85 |
| Site F | 4110 | 1165 | 520 | 1685 | 1906 | 1017 | 2923 | 0.73 | [0.71, 0.76] | 0.74 | [0.71, 0.77] | 0.20 | 0.96 |
| Site G | 16260 | 6414 | 873 | 7287 | 6803 | 889 | 7692 | 0.74 | [0.73, 0.75] | 0.71 | [0.68, 0.75] | 0.02 | 0.75 |
| Site H | 2975 | 0 | 0 | 0 | 1760 | 348 | 2108 | 0.74 | [0.71, 0.77] | 0.75 | [0.68, 0.80] | -0.43 | 0.76 |
| **Total** | 37090 | 9620 | 5040 | 14660 | 14810 | 8393 | 23203 | | | | | | |

## Discussion

We trained a distributed logistic regression model on 14 810 NSCLC patients and validated it on 8 393 patients from eight sites worldwide, yielding a total of 23 203 patients. While we thus easily exceeded the goal of 20 000 by 16.0%, the eight participating sites originate from only five countries which is one country short of the intended goal.

Applying FAIR principles in this project highlighted the challenges in introducing modern data storage and processing approaches in a clinical research context. Semantic web technology allows concepts and relationships between concepts to be coded which makes data more interpretable – an important FAIR principle. The use of semantic web technology requires expertise that is often not present at healthcare institutes. In this project, we worked closely with all partners to support installations. Future projects would benefit from user-friendly software assisting healthcare institutes in transforming their data according to FAIR principles. Creating such software is the goal of an ongoing research project in the CORAL community.



We observed heterogeneity in modelled variables (T, N, M, and overall stage) and outcome (two-year survival) between sites. Sites provided different cohort types, either (complete) clinical records of heterogeneous NSCLC cases or study cohorts with narrower inclusion criteria which can explain much of this heterogeneity (Table 4). Specifically, site A had a biased inclusion towards surviving patients (89.1% two-year survival, Figure 1) and site C provided study cohorts. For both sites, these biases skewed T, N, M, and overall stage distributions towards lower stages. Even for sites providing data based on their full clinical records, different model variable distributions are not surprising since healthcare providers treat different patient subgroups. For example, data in site F originates from a radiotherapy clinic while the data in site G is provided by a comprehensive cancer care center offering different treatments (surgery, (chemo-)radiotherapy, etc.).

For differences in model outcome (two-year survival), there are multiple (possible) causes. For example, site A experienced a biased collection of survival information due to its unavailability in the healthcare provider's Electronic Medical Records (EMR) and the difficulty of retrospectively gathering this missing information when there is no access to survival registries. Furthermore, some sites contributed historical data dating back to 1978 where treatment outcomes were generally worse. Additionally, treatment choices for patient subgroups differ due to national and local treatment guidelines. Another explanation is the difference in patient subgroups admitted for treatment with possibly worse prognosis, e.g., patients with metastasized NSCLC.

Heterogeneity throughout the network is generally advantageous for prediction modelling as it allows models to be trained that are generalizable to a wider range of patients. On the other hand, if the difference in cohorts is caused by characteristics not considered by the model, e.g., difference in treatments or data collection biases, then these differences can have a negative effect on model performance. In our study, site A suffered from a biased inclusion of surviving patients. The effect on the trained model should be low as site A only contributed 7.3% of the training cohort (Figure 2a). However, the usefulness of this dataset for model validation is limited because the performance of this model has not been evaluated for the entire patient population of the site but only for the subgroup following the biased collection (long survivors or recent patients, Table 4). A further inclusion bias is present in site C which provided two study cohorts (predominantly overall stage I and III) for training and validation. Care has to be taken when interpreting validation results: one can only draw conclusions for the patient subpopulation from which the validation dataset has been sampled.

Inter-comparison of summary statistics between sites highlights significant differences in variable distributions that can then be investigated to assure data quality. For example, earlier in this study, the N stage statistics showed one site to have an excess of N3 incidence as compared to other sites. This was subsequently investigated and uncovered a processing error at that site. This role will become increasingly important as outcome modelling studies move away from curated clinical trial datasets and towards routinely collected data and structured information retrospectively extracted from clinical notes.

We also observed varying model performance between sites: the validation cohort AUCs ranged from 0.58 (site D) to 0.85 (site A) and calibration plots (Supplementary Information, Figure S1) display obvious differences. Multiple factors might influence stable performance across sites: e.g., the aforementioned heterogeneity due to unobserved but important variables, or different staging practices across sites. Methods to detect and analyze these discrepancies are yet to be developed. Future work can take advantage of the large patient numbers in the network to analyze subgroups of similar patients to generate better performing models. The Personal Health Train infrastructure provides the means to conduct such analyses.

We observe that our results are qualitatively in accordance with the AJCC TNM cancer staging system: the regression coefficients of the presented model (Table 2) indicate decreased survival probabilities for increases in T, N, M, and overall stage supercategories (with exception of T4). For example, the regression coefficients for overall stage supercategories decrease from 1.05 for overall stage category 0 to -0.82 for overall stage category IV. Additionally, we quantitatively compared the presented model to the AJCC TNM cancer staging system: we retrieved two-year survival probabilities for the overall stages IA, IB, IIA, IIB, IIIA, IIIB, IV of the AJCC TNM cancer staging edition 7[10] (which is the effective edition of the validation cohort) and predicted two-year survival in the validation cohort. Patients with overall stages other than IA, IB, IIA, IIB, IIIA, IIIB, IV were excluded because these stages are either not defined or survival probabilities are not reported in TNM edition 7. AUCs of the presented model and the AJCC TNM cancer staging edition 7 coincided (Supplementary Information, Table S1).

Published NSCLC two-year survival prediction models report AUCs of, for example, 0.68-0.77[11,12]. Comparing the presented model's performance with published models is difficult for multiple reasons:

- inclusion criteria: patient inclusion is restricted to treatment with curative intent[12] or different treatment techniques[13];
- methodology: Cox regression models[14–16] or early mortality[17] predictions are not directly comparable to two-year survival predictions;
- performance estimates: sizes of validation cohorts vary across studies, causing different degrees of variability in the performance estimates, therefore rendering comparison unreliable.

The presented model is trained and validated on patients exhibiting all NSCLC stages, including stage IV patients who are generally not treated with curative intent, have the worst prognosis, and are least likely to survive two years after diagnosis (the two-year survival probability is approximately 10% according to the seventh edition AJCC TNM cancer staging manual[10]). Their bad prognosis is easily predicted but published studies mostly do not include stage IV NSCLC patients. Therefore, the presented model's estimated two-year survival prediction performance is expected to be higher than for published models.

For this project, we have implemented logistic regression, a tool popular in statistical analysis and machine learning for its simplicity and interpretability. Despite logistic regression being

a simple method, penalized logistic regression ranked second in discriminative performance after random forest (which is a much less interpretable classifier) in a recent empirical analysis of six classification algorithms for radiotherapy outcome prediction[18]. The presented model is unpenalized. Penalization might help the individual regression coefficients to converge as it alleviates the multicollinearity problem (Figure 2b) and will be explored in future studies.

With this study, we extend the list of distributed methods that are already implemented in the PHT: Bayesian networks[19] and linear support vector machines[4]. Distributed learning approaches for other machine learning methods are available for future implementation, e.g., (convolutional) neural networks[20].



An alternative to the PHT is DataSHIELD[21], a mature open-source distributed data analysis and machine learning platform with multiple applications. It is based on the open-source software R and Opal data warehouses. The PHT infrastructure differentiates itself from DataSHIELD in multiple aspects:

- it is not limited to R but is compatible with multiple languages (e.g., Java, MATLAB, C#, Python, R);
- it offers analytical flexibility by not limiting the researcher to a fixed function library (DataSHIELD v4.0 comprises 140 R functions[21]);
- it uses Semantic Web technology to store and query data at sites but also allows relational databases and SQL queries;
- the long-term aim of the PHT infrastructure is to connect databases with routine clinical care data.

The presented PHT study only considers a very limited number of clinical data elements (T, N, M, overall stage, diagnosis year, survival follow-up). Arguably, individual predictions need many more data elements. Additional clinical (e.g., age, comorbidities), biological (e.g., genomics, proteomics), imaging (e.g., screening, radiomics[22]) and treatment sources (e.g., radiotherapy treatment planning) are likely to contain relevant data elements for the prediction of a survival outcome. Furthermore, the two-year survival outcome is not sufficient for clinical decision support, quality-of-life, toxicity and cost are also relevant for a balanced decision to be taken. However, due to the limited number of data elements required for inclusion, we could reach very high inclusion numbers and could show that the methodology of distributed learning scales to these numbers. Although the data quality is improving in routine care, the more data elements a study requires, the less complete datasets will be available. As quality improves, future studies are possible where additional data elements (not only prognostic but also predictive for treatment outcomes) can be included and thus better and more clinically relevant models can be developed using the proposed infrastructure.

This project shows distributed learning infrastructures are capable of delivering cohort sizes to rival those available to researchers from national registries. However, distributed approaches such as the PHT, where each institute must only satisfy its local information and research governance requirements, ease the bureaucratic burden of learning from internationally separated pools of patients, particularly between countries with differing information governance regimes. Furthermore, the system is much more flexible and makes including additional data elements into analyses a simple process. If an item is not present in a registry dataset, retrospectively adding this information to previous years is very difficult if not logistically impossible. Lastly, the infrastructure provides a mechanism to expedite the external validation of prognostic and predictive models in cohorts from different countries with different patient demographics, organizational cultures, and treatment regimens.

Changes in the AJCC TNM cancer staging edition were not considered in this study nor was the more granular classification (e.g., T1a, T1b). The staging edition which was used by the physician is not often noted explicitly but future analyses may use the diagnosis year (or institutional information on when they 'switched' editions) as a predictor in the logistic regression model. Knowing the staging edition on a per patient level would make it possible to validate if more recent staging editions are indeed more prognostic and could generally improve the predictive performance of the trained models.

This study has shown that distributed machine learning using Semantic Web technology can be implemented in a short time frame to answer specific research questions. In future work, we will extend the CORAL community with more cancer centers and include more data elements noted in routine care. We expect therefore that, the PHT will enable researchers to rapidly train new prediction models as new patients and data elements become available: accelerating the speed at which clinical observations are turned into actionable knowledge.

The Personal Health Train infrastructure was deployed across eight healthcare institutes in five countries in four months. A two-year survival prediction model was trained and validated in more than 20 000 non-small cell lung cancer patients. This infrastructure demonstrably overcomes patient-privacy barriers to healthcare data sharing and implements distributed data analysis and machine learning across healthcare providers worldwide.

## Methods

This study was registered on clinicaltrials.gov[8] (**https://www.clinicaltrials.gov/ct2/show/NCT03564457**) on 11-06-2018 (first posted date: 20-06-2018, actual study start date: 01-07-2018). In all sites, the project was approved by their institutional review boards (IRBs) or was conform to national information and research governance regulations. Official project invitations were sent to eight sites on 18-06-2018 and two additional sites were contacted later but before the deadline of September 1. Figure 3 shows the project timeline.



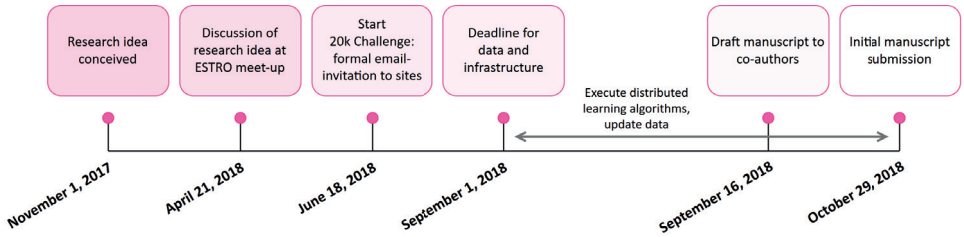



**Figure 3.** Project timeline. ESTRO: European Society for Radiotherapy and Oncology.

Given that the PHT is a privacy-by-design infrastructure where no individual patient data leaves the individual healthcare provider, no researcher has access to the data, data is anonymized or pseudonymized, and given the number of patients involved, internal privacy officers often felt informed consent was neither feasible nor necessary.

### *Patients*

Exports of routine clinical care databases (sites A-B and D-H) or study cohorts (site C) identified as non-small cell lung cancer patients were included in this study (Table 4). Data elements retrieved were the diagnosis, diagnosis date, T, N, and M stages, overall stage, date of last follow-up after the diagnosis date and vital status at last follow-up (alive or dead). If the diagnosis date was not available, date of first treatment, date of histology or date of intake were allowed as a surrogate for the date of diagnosis. Various staging editions (AJCC TNM cancer staging editions 1-8) were published and implemented during the period of treatment.

Two-year survival was defined as a reported time interval between date of diagnosis and date of last follow-up of more than 2 * 365.24 days with a vital status 'alive' at last follow-up or a reported time interval between date of diagnosis and date of death of more than 2 * 365.24 days. Two-year death was defined as date of death less than 730.48 days after the date of diagnosis. Two-year survival was labelled missing if date of diagnosis, date of last follow-up, or vital status at last follow-up were missing. Two-year survival was also defined as missing if the date of last follow-up was earlier than two years after the date of diagnosis and the vital status at last follow-up was 'alive' (right-censored).

**Table 4.** Cohort information. NSCLC: non-small cell lung cancer. SBRT: stereotactic body radiotherapy. RT: radiotherapy. CHART: continuous, hyperfractionated, accelerated radiotherapy.

| | Disease | Interval | Treatment |
|---|---|---|---|
| **Site A** | NSCLC Stage I-IV (histologically confirmed) | January 2008-August 2016 | (Chemo-)radiotherapy, surgery, chemotherapy. Filtered for having last follow-up records in 2018 or documented vital status. |
| **Site B** | NSCLC, Stage I-IV, histo-cytologically confirmed | October 2004-May 2018 | (Chemo-)radiotherapy, chemotherapy, surgery, multimodality treatment. |
| **Site C** | NSCLC, 1) Peripheral stage I, 2) stage III | 1) 2005-2016, 2) 2008-2013 | 1) SBRT only, 2) concurrent (chemo-)radiotherapy, surgery. |
| **Site D** | NSCLC Stage I-IV (either clinical diagnosis or histologically confirmed) | 2004-2017 | Definitive radiotherapy (55Gy in 20 fractions, CHART, concurrent or sequential chemo-radiotherapy or other standard/accepted radical radiotherapy schedules) excluding SBRT or post-surgery adjuvant RT. |
| **Site E** | NSCLC, Stage I-IV (either clinical diagnosis or histologically confirmed) | 1997-2018 | First available T, N, and M staging information of lung cancer patients treated with curative and palliative RT. Includes post-surgery RT, (chemo-)radiotherapy, recurrences. |
| **Site F** | NSCLC, Stage I-IV | 1982-2018 | First available T, N, M, and overall staging information of lung cancer patients treated with curative and palliative RT. Includes post-surgery RT, (chemo-)radiotherapy, recurrences. |
| **Site G** | NSCLC, Stage I-IV | 1955-2018 | First available T, N, M, and overall staging information of lung cancer patients. Includes surgery, (chemo-) radiotherapy. |
| **Site H** | NSCLC, Stage I-IV | 1971-2018 | First available T, N, M, and overall staging information of all lung cancer patients treated with curative and palliative RT. Includes post-surgery RT, (chemo-)radiotherapy, recurrences, SBRT. |

*FAIR data model*

To make data FAIR, a data model has to be agreed upon between parties. As per prior work[23] we have implemented this model using Semantic Web technology. In Figure S2, a graphical representation of the model is shown and on github[24] (**https://github.com/RadiationOncologyOntology/20kChallenge/wiki/Data-model**) the full data model including used classes and properties can be found.

*FAIR data stations*

Creating FAIR data out of clinical information systems generally involved the following tools

- Source systems: these are the clinical systems in which the data elements required for this study were stored
- ETL: software to extract data from source systems, transform data, and load it into a data warehouse
- Data warehouse: a database where data from multiple source systems are combined
- Mapping: transformation from the data warehouse schema to medical ontologies, e.g., the Radiation Oncology Ontology[23] (ROO) or the National Cancer Institute thesaurus[25] (NCIt)
- Graph database: RDF database where data elements are FAIR.

Table 5 shows an overview of the tools used at the various care providers. To support the setup of mapping and graph database software, installation manuals were distributed and remote support was provided.

**Table 5.** Overview of tools used to make data FAIR. EMR: electronic medical records.

| Provider | Amsterdam (NL) | Cardiff (WAL) | Maastricht (NL) | Manchester (ENG) | Nijmegen (NL) | Rome (IT) | Rotterdam (NL) | Shanghai (CN) |
|---|---|---|---|---|---|---|---|---|
| **Source systems** | NKI-AVL Tumour registry | Canisc (Cancer Network Information System Cymru, NHS Wales Information Services) | HiX (Chipsoft, Netherlands), municipality population registry (survival data) | Clinical Web Portal (in house e-records system). Mosaiq radiotherapy oncology information system. Medway Sigma BI patient administration system. | Radiotherapieweb (in-house EMR), municipality population registry (survival data) | BOA26 and Speed RO | OpenClinica, Microsoft Access | Chinese EMR |
| **ETL tools** | MS SSIS | MATLAB | SAP Business Objects, MATLAB | Pentaho data integration, SQL, Java, Python, R | PHP, SQL, MATLAB | SQL | MATLAB | In-house software |
| **Data warehouse** | MS SQL Server | MS SQL Server | SAP Business Objects | PostgreSQL | SQL Server | SQL Server | | None |
| **Mapping** | D2RQ | | | | | | | |
| **Graph database** | Blazegraph | | | | | | | |

*Network for secure application distribution, execution, and communication*

For the secure distribution of and messaging between applications, a solution called the Varian Learning Portal (VLP, Varian Medical Systems, Palo Alto, CA) was used. The VLP is a cloud-based system which has implemented user, site, and project management so that a research project consisting of multiple data providers and researchers can securely share applications and communication between applications. To connect the VLP to a local data station, a learning connector is installed at each data provider. The learning connector is a gateway through which applications and communication are handled. The iterative execution of applications and communication between them is called a learning run and each data provider can accept or deny each learning run. All communication and other actions are logged and auditable by members of a given project.

*Applications for distributed cohort discovery, and learning*

The VLP allows a certificate-based upload of applications. Each application group has two parts. One that runs at the VLP in the cloud (master application) and one at each of the sites (site application). Multiple application groups were developed in this project.

- The first application group's aim is cohort discovery. An application is sent to each site to determine and communicate generic statistics (counts) of the available data in the FAIR data station. This cohort discovery application includes a SPARQL Protocol and RDF Query Language (SPARQL) query. Each site application reports its site statistics to a master application running at the VLP which are then reported back to the researcher who initiated the application. Multiple variations of this application group were employed to generate summary statistics for patient subgroups.
- The second application group aims to train a logistic regression (LR) model. Each LR site application can, given a SPARQL query, train a LR model from the local dataset. The regression coefficients of each site LR model and patient counts are then sent to the master application that reaches consensus in an iterative manner. Figure 4 illustrates the process followed in the LR application group.
- The third application group validates a given LR model on the sites. An application is sent to each site to compute model performance metrics (RMSE, ROC curve, AUC, calibration plots) and transfers these back to the master application which combines and passes them on to the researcher. Calibration plots reporting calibration-in-the-large and calibration slope are generated following Steyerberg[27] and include Wilson confidence intervals implemented by Winkler and Nichols[28].

The LR model is trained on patients treated between 1978 and 2012 and validated on all patients treated between 2012 and 2015. Only patients with complete diagnosis date, follow-up date, follow-up status, and complete T, N, M, and overall stage after imputation are included. This approach simulates the development of an LR model and sequential validation on new data becoming available over time. This is a TRIPOD type 2b validation[9].

The application used to train the LR coefficients in a distributed manner is based on the Alternating Direction Method of Multipliers (ADMM) and exemplary implementations by Boyd et al.(2011)[29,30]. ADMM decomposes the optimization problem underlying logistic regression (finding regression coefficients that maximize the log-likelihood of all training data) into an iterative optimization: each site computes regression coefficients that optimize a trade-off between maximizing the log-likelihood for the site's local data and a degree of agreement with the network consensus (a combination of the regression coefficients determined at all sites). This trade-off includes a penalty for disagreeing with this consensus.

At the master, the sets of site-specific regression coefficients are combined to a new consensus and a new disagreement penalty value is determined. This consensus and the new penalty are then returned to each site to again optimize site-specific coefficients (the trade-off between maximizing log-likelihood and agreement with consensus changes because of the new consensus and disagreement penalty). This iterative procedure is repeated until the discrepancy between the sites' local coefficients and the consensus, as well as the change in the consensus solutions over iterations is sufficiently small. For an excellent technical description of ADMM, we suggest Boyd et al. (2011)[29]. All application groups are implemented in MATLAB R2018a (Mathworks, Natick, MA). Code and accompanying documentation are available open-source[31] (**https://github.com/RadiationOncologyOntology/20kChallenge**).



*Data processing before LR training*

The levels for each variable (T, N, M, and overall stage) are grouped in supercategories (Table 6) to allow regression on data of different AJCC TNM cancer staging editions and to bundle similar categories.

**Table 6.** Supercategories for T, N, M, and overall stages grouping AJCC TNM cancer staging editions 1-8.

| T | | N | | M | | Overall stage | |
|---|---|---|---|---|---|---|---|
| **0** | T0 | **0** | N0 | **0** | M0 | **0** | 0 |
| **1** | T1, T1a, T1b, T1c, T1mi, Tis | **1** | N1 | **1** | M1, M1a, M1b, M1c | **I** | IA, IA1, IA2, IA3, IB |
| **2** | T2, T2a, T2b | **2** | N2 | **X** | MX | **II** | IIA, IIB |
| **3** | T3 | **3** | N3 | | | **III** | III, IIIA, IIIB, IIIC |
| **4** | T4 | **X** | NX | | | **IV** | IV, IVA, IVB |
| **X** | TX | | | | | **Occult** | Occult |

T, N, M, and overall stages were dummy-coded to estimate the individual effect of each stage on two-year survival. T1, N0, M0 and overall stage I categories were used as the reference categories to avoid multicollinearity issues in the regression model. For example, the ordinal variable T stage, which takes six values (0 to 4, X), is converted to five binary variables representing T0, T2, T3, T4, TX.

*Imputation*

If a patient misses entries for one or more of the variables T, N, M, and/or overall staging (but not all of them), imputation of the missing values is attempted. A detailed imputation process description is presented in Figure S3 (Supplementary Information) and an outline is given below.

First, the missing values are logically induced from the permitted combinations of T, N, M, and overall stages. For example, a patient diagnosed in 2011 with N0M0 and overall stage IIA but missing T can only have T2b according to TNM edition 7.

If the logical imputation is ambiguous because multiple imputation results are possible, the missing values are imputed probabilistically based on a subset of patients treated at the same site. This subset contains patients treated at the same site, within the time interval corresponding to the selected AJCC TNM edition, and matching the available variables of the patient. This subset also contains patients for which missing values are logically imputed so that probabilistic imputation is also feasible for sites E and H which miss some variable for all patients. The empirical probability of each T, N, M, and overall stage combination observed in this patient subset is computed and one of these combinations is randomly sampled according to the computed empirical probabilities. For example, a patient diagnosed in 2013 with T1aN0 and overall stage IV but missing M can be imputed with M1a or M1b according to TNM edition 7. If there are 30 patients with T1aN0M1a & overall stage IV and 70 patients with T1aN0M1b & overall stage IV diagnosed starting 2010 and before 2018, the missing M value is imputed by 1a with probability 0.3 and by 1b with probability 0.7. This probabilistic imputation procedure assumes variables to be missing at random which is a simplifying assumption in routine clinical care data.

This imputation procedure is repeated for all available TNM editions (1-8). To decide on a single imputation for a given patient, the most recent TNM edition meeting two criteria is selected:

- it was effective before or in the patient's year of diagnosis;
- it yields a complete imputation.

The modeling choice to also use preceding editions takes into account the possibility that the treating physician has not yet adopted the newest AJCC TNM cancer staging edition.

The following official effective dates for AJCC TNM cancer staging editions are used[32]:

- Edition 1: 1978 – 1983
- Edition 2: 1984 – 1988
- Edition 3: 1989 – 1992
- Edition 4: 1993 – 1997
- Edition 5: 1998 – 2002
- Edition 6: 2003 – 2009
- Edition 7: 2010 – 2017
- Edition 8: 2018 – present

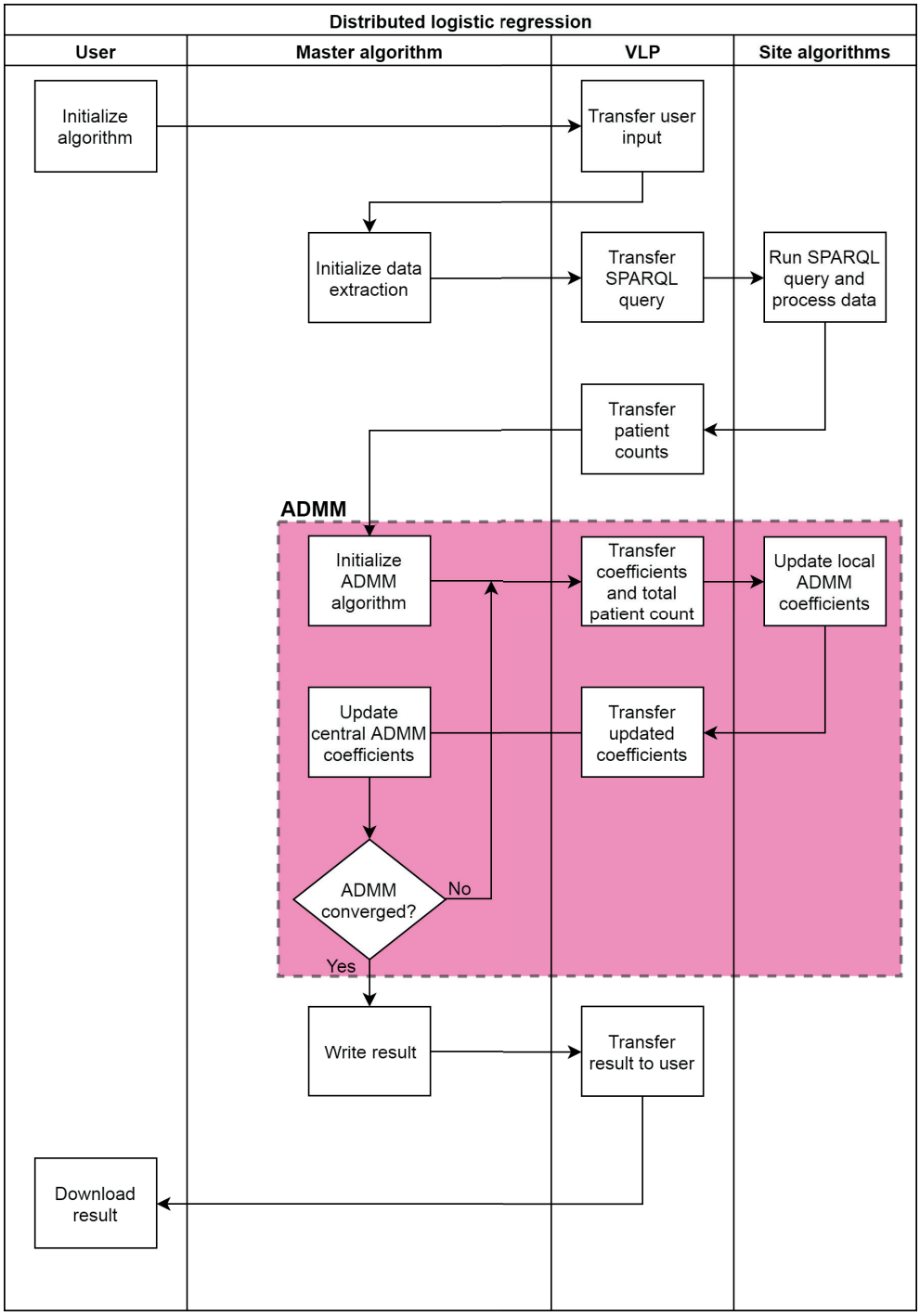



**Figure 4.** A simplified process description of the distributed logistic regression application group. VLP: Varian Learning Portal. ADMM: Alternating Direction Method of Multipliers.

**Acknowledgements**
We would like to thank Wolfgang Wiessler (Varian Medical Systems) for his advice and technical support. Sophie Stovold is acknowledged for her work in developing the Velindre (Cardiff) database. Mieke Basten and Thierry Felkers are acknowledged for their work in developing the Nijmegen database. Els Berenschot-Huijbregts and Andras Zolnay are acknowledged for their work in developing the Rotterdam database. Robbert Hardenberg and Tony van de Velde are acknowledged for their work in developing the Amsterdam database.

We would like to thank the following colleagues of the MDTB:

- Giovanna Mantini,[6,7] Department Radiation Oncology
- A. Martino,[7] Department Radiation Oncology
- L. Boldrini,[6,7] Department Radiation Oncology
- A. Damiani,[7] Department Radiation Oncology
- S. Margaritora,[6,7] Department of Surgery
- M.T. Cogedo,[7] Department of Surgery
- F. Lococo,[7] Department of Surgery
- A. Farchione[7] Department Radiology
- G. Rindi,[6,7] Department of Pathology

We wish to acknowledge technical and financial support from the following organizations: Varian Medical Systems (VLP, SAGE); Netherlands Organisation for Scientific Research (grant n° 10696 DuCAT, BIONIC, VWData); Province of Limburg (LIME); Dutch Cancer Society (TraIT2HealthRI, PROTRAIT); Health-RI; Netherlands Federation of University Medical Centres (Data4LifeSciences). This research is also supported by ERC advanced grant (ERC-ADG-2015, n° 694812), EUROSTARS (DART, DECIDE), the European Program H2020-2015-17 ImmunoSABR - n° 733008, PREDICT - ITN - n° 766276, TRANSCAN JointTransnational Call 2016 (JTC2016 "CLEARLY"- n° UM 2017-8295), Interreg V-A Euregio Meuse-Rhine ("Euradiomics") and Kankeronderzoekfonds Limburg from the Health Foundation Limburg; Cardiff University Data Innovation Research Institute Seedcorn Fund grant n° 23020-AC23024072/16; Velindre NHS Trust Charitable Funds grant n° 2017/12.

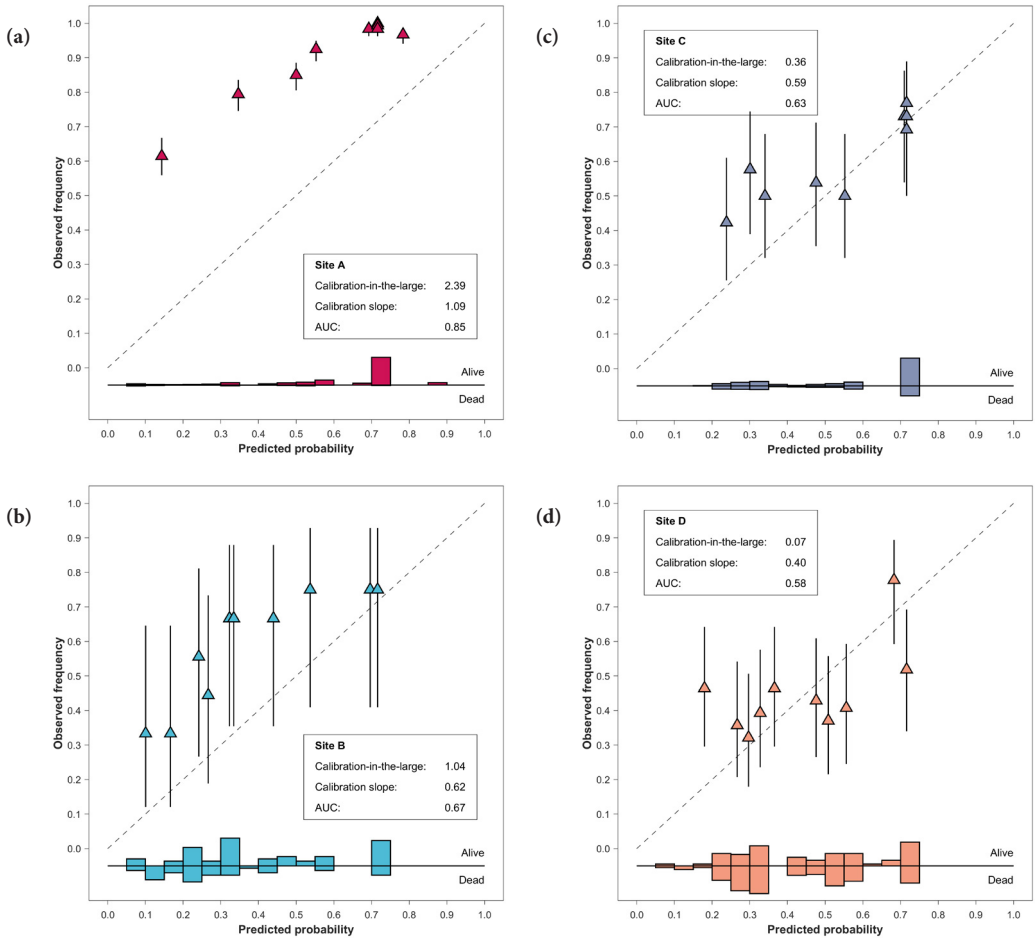
Gareth Price and Corinne Faivre-Finn acknowledge the support of Cancer Research UK via funding to the Cancer Research Manchester Centre [C147/A18083] and [C147/A25254].

## References


1. Sullivan, R. *et al.* Delivering affordable cancer care in high-income countries. *Lancet Oncol.* **12**, 933–980 (2011).
2. Personal Health Train. *Dutch Techcentre for Life Sciences* Available at: https://www.dtls.nl/fair-data/personal-health-train/. (Accessed: 12th September 2018)
3. Wilkinson, M. D. *et al.* The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* (2016). doi:10.1038/sdata.2016.18
4. Deist, T. M. *et al.* Infrastructure and distributed learning methodology for privacy-preserving multi-centric rapid learning health care: euroCAT. *Clin. Transl. Radiat. Oncol.* **4**, 24–31 (2017).
5. Lambin, P. *et al.* 'Rapid Learning health care in oncology' - an approach towards decision support systems enabling customised radiotherapy'. *Radiother. Oncol. J. Eur. Soc. Ther. Radiol. Oncol.* **109**, 159–64 (2013).
6. Gaye, A. *et al.* DataSHIELD: taking the analysis to the data, not the data to the analysis. *Int. J. Epidemiol.* **43**, 1929–1944 (2014).
7. Hripcsak, G. *et al.* Observational Health Data Sciences and Informatics (OHDSI): Opportunities for Observational Researchers. *Stud. Health Technol. Inform.* **216**, 574–8 (2015).
8. 20K Distributed Learning Challenge - ClinicalTrials.gov. Available at: https://clinicaltrials.gov/ct2/show/NCT03564457. (Accessed: 12th September 2018)
9. Collins, G. S., Reitsma, J. B., Altman, D. G. & Moons, K. G. Transparent reporting of a multivariable prediction model for individual prognosis or diagnosis (TRIPOD): the TRIPOD Statement. *BMC Med.* **13**, 1 (2015).
10. *AJCC cancer staging manual*. (Springer, 2010).
11. Jayasurya, K. *et al.* Comparison of Bayesian network and support vector machine models for two-year survival prediction in lung cancer patients treated with radiotherapy. *Med. Phys.* **37**, 1401–1407 (2010).
12. Dehing-Oberije, C. *et al.* Development and external validation of prognostic model for 2-year survival of non-small-cell lung cancer patients treated with chemoradiotherapy. *Int. J. Radiat. Oncol. Biol. Phys.* **74**, 355–362 (2009).
13. Mao, Q. *et al.* A nomogram to predict the survival of stage IIIA-N2 non-small cell lung cancer after surgery. *J. Thorac. Cardiovasc. Surg.* **155**, 1784-1792.e3 (2018).
14. Carvalho, S. *et al.* Prognostic value of blood-biomarkers related to hypoxia, inflammation, immune response and tumour load in non-small cell lung cancer – A survival model with external validation. *Radiother. Oncol.* **119**, 487–494 (2016).
15. Oberije, C. *et al.* A Validated Prediction Model for Overall Survival From Stage III Non-Small Cell Lung Cancer: Toward Survival Prediction for Individual Patients. *Int. J. Radiat. Oncol. Biol. Phys.* **92**, 935–944 (2015).
16. Chaddad, A., Desrosiers, C., Toews, M. & Abdulkarim, B. Predicting survival time of lung cancer patients using radiomic analysis. *Oncotarget* **8**, 104393–104407 (2017).
17. Jochems, A. *et al.* A prediction model for early death in non-small cell lung cancer patients following curative-intent chemoradiotherapy. *Acta Oncol. Stockh. Swed.* **57**, 226–230 (2018).
18. Deist, T. M. *et al.* Machine learning algorithms for outcome prediction in (chemo)radiotherapy: An empirical comparison of classifiers. *Med. Phys.* **45**, 3449–3459 (2018).
19. Jochems, A. *et al.* Distributed learning: Developing a predictive model based on data from multiple hospitals without data leaving the hospital – A real life proof of concept. *Radiother. Oncol.* **121**, 459–467 (2016).

20. McMahan, H. B., Moore, E., Ramage, D., Hampson, S. & Arcas, B. A. y. Communication-Efficient Learning of Deep Networks from Decentralized Data. *ArXiv160205629 Cs* (2016).
21. Wilson, R. C. *et al.* DataSHIELD – new directions and dimensions. *Data Sci. J.* **16**, 1–21 (2017).
22. Aerts, H. J. W. L. *et al.* Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach. *Nat Commun* **5**, (2014).
23. Traverso, A., Soest, J. van, Wee, L. & Dekker, A. The radiation oncology ontology (ROO): Publishing linked data in radiation oncology using semantic web and ontology techniques. *Med. Phys.* (accepted). doi:10.1002/mp.12879
24. Distributed learning over 20k+ patients. (2018). Available at: https://github.com/RadiationOncologyOntology/20kChallenge. (Accessed: 12th September 2018)
25. NCI Thesaurus. Available at: https://ncit.nci.nih.gov/ncitbrowser/. (Accessed: 12th September 2018)
26. Tagliaferri, L. *et al.* ENT COBRA (Consortium for Brachytherapy Data Analysis): interdisciplinary standardized data collection system for head and neck patients treated with interventional radiotherapy (brachytherapy). *J. Contemp. Brachytherapy* **8**, 336–343 (2016).
27. Steyerberg, E. W. Evaluation of performance. in *Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating* (ed. Steyerberg, E. W.) 255–280 (Springer New York, 2009). doi:10.1007/978-0-387-77244-8_15
28. Winkler, A. M. Confidence intervals for Bernoulli trials. *Brainder.* (2012).
29. Boyd, S., Parikh, N., Chu, E., Peleato, B. & Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **3**, 1–122 (2011).
30. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. Available at: http://web.stanford.edu/~boyd/papers/admm_distr_stats.html. (Accessed: 3rd October 2018)
31. *Distributed learning over 20k+ patients. Contribute to RadiationOncologyOntology/20kChallenge development by creating an account on GitHub*. (RadiationOncologyOntology, 2018).
32. AJCC - Cancer Staging Manual. Available at: http://cancerstaging.org/references-tools/deskreferences/Pages/default.aspx. (Accessed: 12th September 2018)

## Supplementary Information



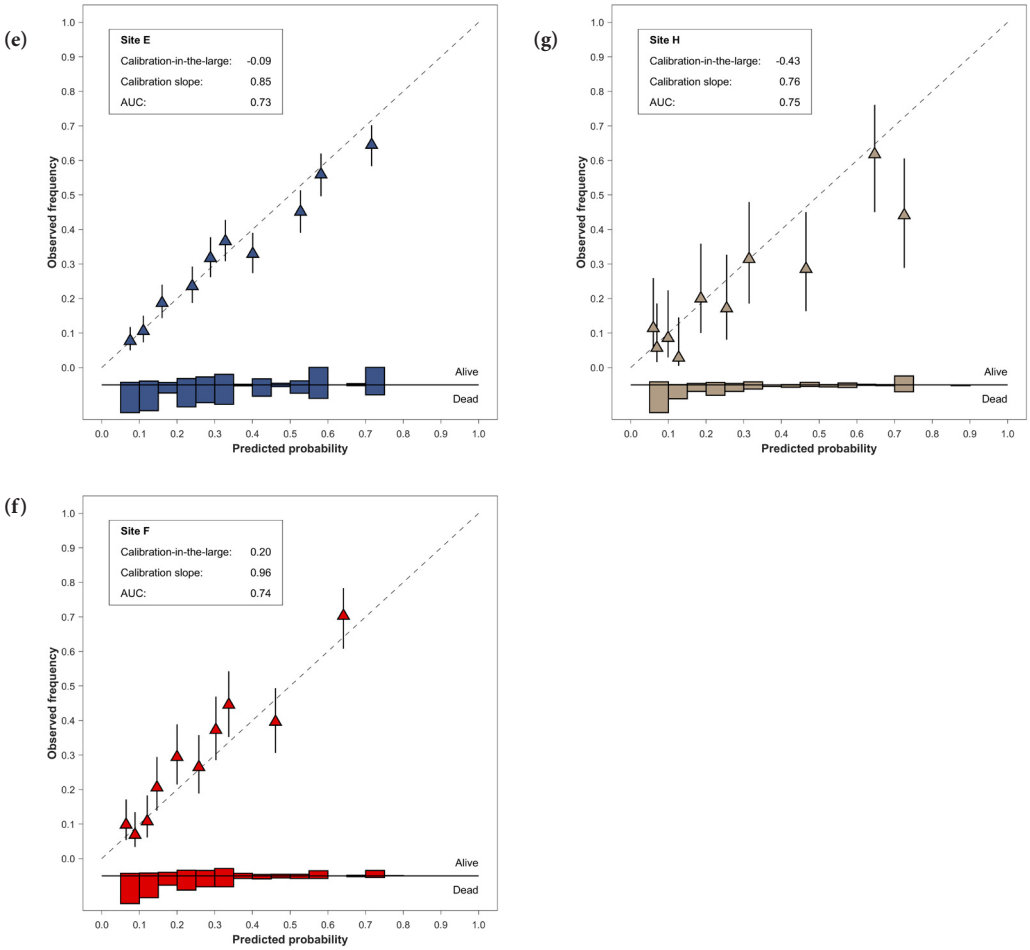



**Figure S1.** Calibration plots of the validation data for all sites excluding site G (Figure 2d). AUC: area under the receiver operating characteristic curve.

**Table S1.** Patient counts with stages IA, IB, IIA, IIB, IIIA, IIIB, IV in the validation cohort and corresponding model performance per site for the presented model and the AJCC TNM cancer staging edition 7 survival probabilities[10]. Survival probabilities for stage 0 and Occult are not available in the reference. The corresponding patients were thus excluded. Patients not staged according to edition 7 in the validation cohort were also excluded. AUC: area under the receiver operating characteristic curve. CI: confidence interval.

| Site | Validation cohort patient counts (stage IA, IB, IIA, IIB, IIIA, IIIB, IV) | Model performance | | | | |
|---|---|---|---|---|---|---|
| | | Logistic regression | | AJCC edition 7 | | |
| | | AUC | 95%-CI | AUC | 95%-CI | Δ AUC |
| Site A | 2803 | 0.87 | [0.84, 0.89] | 0.86 | [0.84, 0.89] | 0.00 |
| Site B | 87 | 0.67 | [0.54, 0.78] | 0.69 | [0.56, 0.80] | -0.02 |
| Site C | 131 | 0.54 | [0.43, 0.64] | 0.52 | [0.42, 0.61] | 0.02 |
| Site D | 273 | 0.59 | [0.52, 0.66] | 0.59 | [0.53, 0.65] | 0.00 |
| Site E | 2455 | 0.73 | [0.70, 0.74] | 0.71 | [0.69, 0.73] | 0.01 |
| Site F | 939 | 0.73 | [0.69, 0.76] | 0.72 | [0.68, 0.75] | 0.01 |
| Site G | 878 | 0.71 | [0.67, 0.75] | 0.71 | [0.66, 0.74] | 0.01 |
| Site H | 341 | 0.76 | [0.69, 0.82] | 0.77 | [0.69, 0.81] | -0.01 |
| **Total** | 7907 | | | | | |

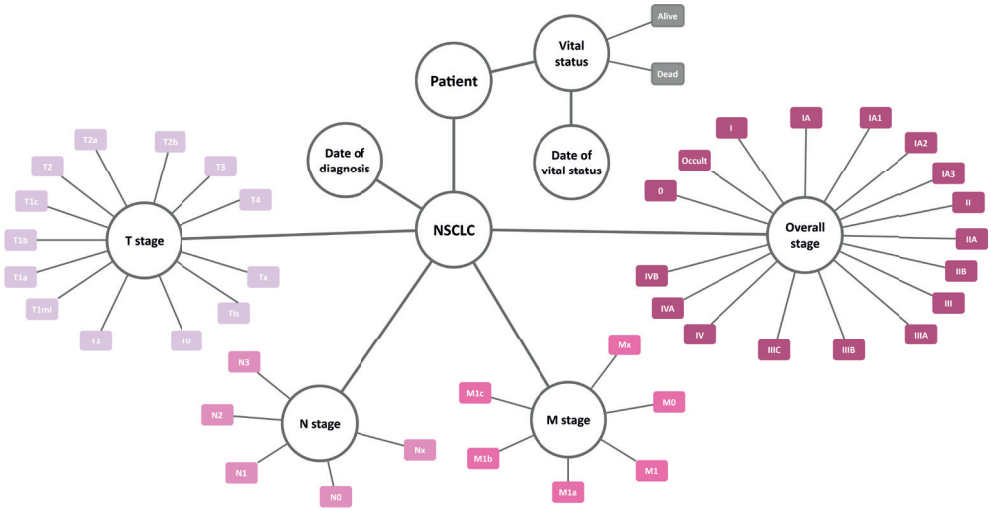
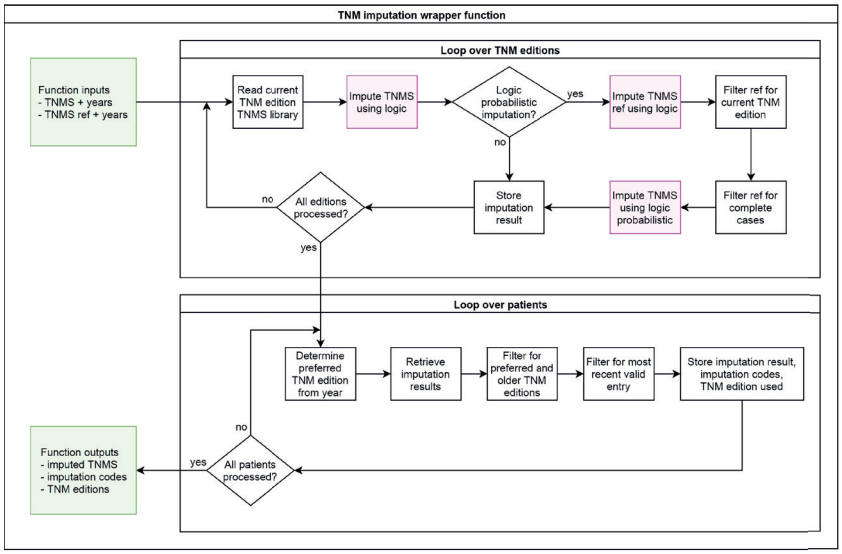
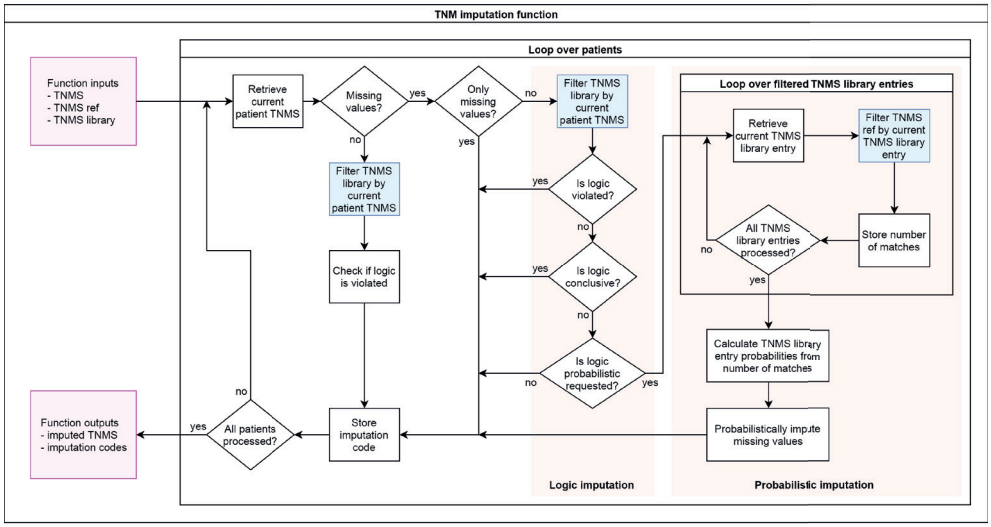



**Figure S2.** A graphical representation of the data model employed in the distributed learning network.

(a)


TNM imputation wrapper function
Loop over TNM editions
Function inputs
- TNMS + years
- TNMS ref + years
Read current TNM edition TNMS library
Impute TNMS using logic
Logic probabilistic imputation?
yes
no
Impute TNMS ref using logic
Filter ref for current TNM edition
Filter ref for complete cases
Impute TNMS using logic probabilistic
Store imputation result
All editions processed?
no
yes
Loop over patients
Determine preferred TNM edition from year
Retrieve imputation results
Filter for preferred and older TNM editions
Filter for most recent valid entry
Store imputation result, imputation codes, TNM edition used
All patients processed?
no
yes
Function outputs
- imputed TNMS
- imputation codes
- TNM editions

(b)


TNM imputation function
Loop over patients
Function inputs
- TNMS
- TNMS ref
- TNMS library
Retrieve current patient TNMS
Missing values?
yes
no
Only missing values?
no
yes
Filter TNMS library by current patient TNMS
Filter TNMS library by current patient TNMS
Is logic violated?
yes
no
Check if logic is violated
Is logic conclusive?
yes
no
Is logic probabilistic requested?
no
yes
Loop over filtered TNMS library entries
Retrieve current TNMS library entry
Filter TNMS ref by current TNMS library entry
All TNMS library entries processed?
no
yes
Store number of matches
Calculate TNMS library entry probabilities from number of matches
Probabilistically impute missing values
Store imputation code
All patients processed?
no
yes
Function outputs
- imputed TNMS
- imputation codes
Logic imputation
Probabilistic imputation

(c)

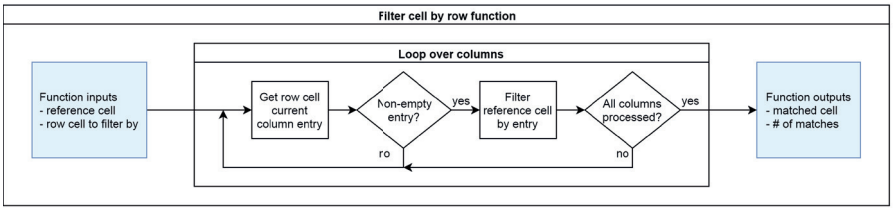



**Figure S3.** Imputation process. The TNM imputation wrapper function (a) is the outermost function which uses the TNM imputation function (b) and the Filter cell by row function (c) as subfunctions. The wrapper function has two input groups: data for the patients that are to be imputed and data for patients that act as the reference for probabilistic imputation. For both input groups, T, N, M, overall stage, and diagnosis year per patient are needed. TNM: cancer staging system based on tumor size (T), lymph node involvement (N) and metastasis (M). TNM edition: one of eight released TNM cancer staging system editions effective since 1978 and in non-overlapping time periods. TNMS: combination of TNM and cancer stage (S) for a patient (can contain missing values) or in the TNMS library (complete cases). Years: year of diagnosis corresponding with time of TNM staging, and used to determine the currently effective TNM edition. TNMS ref: reference patient TNMS combinations to be used for logic probabilistic imputation. TNMS library: library of valid combinations of TNM and cancer stages according to a specific TNM edition. Logic imputation: imputation of a missing TNMS value according to a single conclusive combination in the TNMS library. Logic probabilistic imputation: imputation of a missing TNMS value according to multiple inconclusive combinations in the TNMS library and their respective probabilities of occurrence in the TNMS reference cell. Imputation code: patient specific codes to indicate if the TNMS entries follow the TNM edition logic and the type of imputation performed (if any).

# Centralized learning

# Chapter 4

## Machine learning algorithms for outcome prediction in (chemo)radiotherapy: an empirical comparison of classifiers

Timo M. Deist, Frank J.W.M. Dankers, Gilmer Valdes, Robin Wijsman, I-Chow Hsu, Cary Oberije, Tim Lustberg, Johan van Soest, Frank Hoebers, Arthur Jochems, Issam El Naqa, Leonard Wee, Olivier Morin, David R. Raleigh, Wouter Bots, Johannes H. Kaanders, José Belderbos, Margriet Kwint, Timothy Solberg, René Monshouwer, Johan Bussink, Andre Dekker, Philippe Lambin

***Adapted from*** *Deist, Timo M., et al. "Machine learning algorithms for outcome prediction in (chemo) radiotherapy: An empirical comparison of classifiers." Medical physics (2018).*

## Abstract

### Purpose
Machine learning classification algorithms (classifiers) for prediction of treatment response are becoming more popular in radiotherapy literature. General machine learning literature provides evidence in favor of some classifier families (random forest, support vector machine, gradient boosting) in terms of classification performance. The purpose of this study is to compare such classifiers specifically for (chemo)radiotherapy datasets and to estimate their average discriminative performance for radiation treatment outcome prediction.

### Methods
We collected 12 datasets (3484 patients) from prior studies on post-(chemo)radiotherapy toxicity, survival, or tumor control with clinical, dosimetric, or blood biomarker features from multiple institutions and for different tumor sites, i.e. (non-)small cell lung cancer, head and neck cancer, and meningioma. Six common classification algorithms with built-in feature selection (decision tree, random forest, neural network, support vector machine, elastic net logistic regression, LogitBoost) were applied on each dataset using the popular open-source *R* package *caret*. The *R* code and documentation for the analysis are available online[1]. All classifiers were run on each dataset in a 100-repeated nested 5-fold cross-validation with hyperparameter tuning. Performance metrics (AUC, calibration slope and intercept, accuracy, Cohen's kappa, and Brier score) were computed. We ranked classifiers by AUC to determine which classifier is likely to also perform well in future studies. We simulated the benefit for potential investigators to select a certain classifier for a new dataset based on our study (*pre-selection* based on other datasets) or estimating the best classifier for a dataset (*set-specific selection* based on information from the new dataset) compared to uninformed classifier selection (random selection).

### Results
Random forest (best in 6/12 datasets) and elastic net logistic regression (best in 4/12 datasets) showed the overall best discrimination but there was no single best classifier across datasets. Both classifiers had a median AUC *rank* of 2. Pre-selection and set-specific selection yielded a significant average AUC improvement of 0.02 and 0.02 over random selection with an average AUC *rank* improvement of 0.52 and 0.65, respectively.

### Conclusion
Random forest and elastic net logistic regression yield higher discriminative performance in (chemo)radiotherapy outcome and toxicity prediction than other studied classifiers. Thus, one of these two classifiers should be the first choice for investigators when building classification models or to benchmark one's own modelling results against. Our results also show that an informed pre-selection of classifiers based on existing datasets can improve discrimination over random selection.

## Introduction

Machine learning algorithms for predicting (chemo)radiotherapy outcomes (e.g., survival, treatment failure, toxicity) are receiving much attention in literature, for example in decision support systems for precision medicine[2,3]. Currently, there is no consensus on an optimal classification algorithm. Investigators select algorithms for various reasons: the investigator's experience, usage in literature, data characteristics and quality, hypothesized feature dependencies, availability of simple implementations, and model interpretability. One objective criterion for selecting a classifier is to maximize a chosen performance metric, e.g., discrimination (expressed by the area under the receiver operating characteristic curve, AUC). Here, we discuss the performance of binary classifiers in (chemo)radiotherapy outcome prediction, i.e. algorithms that predict whether or not a patient has a certain outcome. We empirically study the behaviour of existing simple implementations of classifiers on a range of (chemo)radiotherapy outcome datasets to possibly identify a classifier with overall maximal discriminative performance. This is a relevant question for investigators who search for a rational basis to support their choice of a classifier or who would like to compare their own modelling results to established algorithms. We employ various open-source *R* packages interfaced with the *R* package *caret*[4] (version 6.0-73) that is readily available for investigators and has shown to produce competitive results[5]. With our results, we also wish to provide guidance in the current trend to delegate modelling decisions to machine learning algorithms.



Large scale studies in the general machine learning literature[5–7] provide evidence in favor of some classifier families (random forest (*rf*), support vector machine (*svm*), gradient boosting machine (*gbm*)) in terms of classification performance. In our study, we investigate how these results translate to (chemo)radiotherapy datasets for treatment outcome prediction/prognosis. To the best of our knowledge, this is the first study to investigate classifier performance on a wide range of such datasets. The studied features are clinical, dosimetric, and blood biomarkers.

Within the framework of existing classifier implementations, we attempt to answer three research questions:

- Is there a superior classifier for predictive modelling in (chemo)radiotherapy?
- How dataset-dependent is the choice of a classifier?
- Is there a benefit of choosing a classifier based on empirical evidence from similar datasets (*pre-selection*)?

Parmar et al. (2015)[8] compared multiple classifiers and feature selection methods (i.e. *filter*-based feature selection) on *radiomics* data using the *caret* package. We build upon this work and extend the analysis to 12 datasets outside the *radiomics* domain. We omit *filter* methods because all classifiers in our study comprise built-in feature selection methods (i.e. *embedded* feature selection) and the main advantage of *filter* methods, i.e. low computational cost per feature, is not relevant for our datasets with only modest numbers of features.

## Material and Methods

*Data collection*

Twelve datasets (3484 patients) with treatment outcomes described in previous studies were collected from public repositories (**www.cancerdata.org**) or provided by collaborators. Table 1 characterizes these datasets. Given availability, some datasets consist of subsamples of or contain fewer/more patients and/or features than the cohorts described in the original studies. Two datasets were excluded after a preliminary analysis (these datasets are also not mentioned in table 1) where none of the studied classifiers resulted in an average AUC above 0.51, which is evidence that they contain no discriminative power. Datasets without discriminative power are not suitable for this analysis as we would be unable to determine differences in discriminative performance across classifiers. The patient cohorts of 2 datasets, Wijsman et al. (2015 and 2017), partially overlap but each dataset lists a different outcome (esophagitis and pneumonitis). Datasets were anonymized in the analysis because their identity is not relevant for interpreting the results and to encourage investigators to share their datasets.

Non-binary outcomes were dichotomized, e.g., overall survival was translated into 2-year overall survival in the dataset of Carvalho et al. (2016). Missing data was imputed for training and test sets (the splitting of datasets into training and test sets is described in section *Experimental Design*) by medians for continuous features and modes for categorical features based on the training set. Basing the imputation on the training set avoids information leakage from test to training sets. Categorical features in training and test sets were dummy coded, i.e. representing categorical features as a combination of binary features, based on the combined set for classifiers that cannot handle categorical features (see table 2). Dummy coding on the combined set ensures that the coding represents all values observed in a dataset. Features with zero variance in training sets were deleted in the training set and in the corresponding test set. Additionally, we removed near-zero variance features for *glmnet* to avoid the classifier implementation from crashing during the fitting process. Features in training sets were rescaled to the interval [0,1] and the same transformation was applied to the corresponding test sets. Rescaling is needed for certain classifiers, e.g., *svmRadial*. All these operations (imputation, dummy coding, deleting (near-)zero variance features, rescaling) were performed independently for each pair of training and test sets (step 2 in figure 1).

**Table 1.** Dataset characteristics. The number of features is determined before pre-processing.

| Dataset | Disease | Outcome | Prevalence (in %) | Patients | Features | Feature types | Source |
|---|---|---|---|---|---|---|---|
| Belderbos et al. (2005)[9] | Non-small cell lung cancer | Grade ≥2 acute esophagitis | 27 | 156 | 22 | Clinical, dosimetric, blood | Private |
| Bots et al. (2017)[10] | Head and neck cancer | 2-year overall survival | 42 | 137 | 10 | Clinical, dosimetric | Private |
| Carvalho et al. (2016)[11] | Non-small cell lung cancer | 2-year overall survival | 40 | 363 | 18 | Clinical, dosimetric, blood | Public[12] |
| Janssens et al. (2012)[13] | Laryngeal cancer | 5-year regional control | 89 | 179 | 48 | Clinical, dosimetric, blood | Private |
| Jochems et al. (2016)[14] | Non-small cell lung cancer | 2-year overall survival | 36 | 327 | 9 | Clinical, dosimetric | Private |
| Kwint et al. (2012)[15] | Non-small cell lung cancer | Grade ≥2 acute esophagitis | 61 | 139 | 83 | Clinical, dosimetric, blood | Private |
| Lustberg et al. (2016)[16,17] | Laryngeal cancer | 2-year overall survival | 83 | 922 | 7 | Clinical, dosimetric, blood | Private |
| Morin et al. (forthcoming) | Meningioma | Local failure | 36 | 257 | 18 | Clinical | Private |
| Oberije et al. (2015)[18] | Non-small cell lung cancer | 2-year overall survival | 36 | 536 | 20 | Clinical, dosimetric | Public[19] |
| Olling et al. (2017)[20] | Small and non-small cell lung cancer | Odynophagia prescription medication | 67 | 131 | 47 | Clinical, dosimetric | Private |
| Wijsman et al. (2015)[21] | Non-small cell lung cancer | Grade ≥2 acute esophagitis | 36 | 149 | 11 | Clinical, dosimetric, blood | Private |
| Wijsman et al. (2017)[22] | Non-small cell lung cancer | Grade ≥3 radiation pneumonitis | 14 | 188 | 18 | Clinical, dosimetric, blood | Private |

*Classifiers*

Six common classifiers were selected and their implementations were used via their interfacing with the open-source *R* package *caret*. The selection includes classifiers frequently used in medical data analysis and advanced classifiers such as random forests or neural networks.

- Elastic net logistic regression is a regularized form of logistic regression, which models additive linear effects. The added shrinkage regularization (i.e. feature selection) makes it is suitable for datasets with many features while maintaining the interpretability of a standard logistic regression.
- Random forests generate a large number of decision trees based on random subsamples of the training set while also randomly varying the features used in the trees. Random forests allow modelling non-linear effects. A random forest model is an ensemble of many decision tree models and is therefore difficult to interpret.
- Single-hidden-layer neural networks are simple versions of multi-layer perceptron neural network models, which are currently popularized by deep neural network applications in machine learning. In the hidden layer, auxiliary features are generated from the input features which are then used for classification. The weights used to generate auxiliary features are derived from the training set. The high number of weights require more training data than other simpler algorithms and reduce interpretability. However, if sufficient data is available, complex relationships between features can be modelled.
- Support vector machines with a radial basis function (RBF) kernel transform the original feature space to attain a better separation between classes. This transformation, however, is less intuitive than linear SVMs where a separating hyperplane is in the original feature space.
- LogitBoost (if used with decision stumps as in this paper) learns a linear combination of multiple single feature classifiers. Training samples that are misclassified in early iterations of the algorithm are given a higher weight when determining further classifiers. The final model is a weighted sum of single feature classifiers. Similar to random forests, it builds an ensemble of models which is difficult to interpret.
- A decision tree iteratively subdivides the training set by selecting feature cutoffs. Decision trees can model non-linear effects and are easily interpretable as long as the tree depth is low.

Classifier details can be found in general machine learning textbooks[23,24]. Table 2 further characterizes these classifiers. We use the option in *caret* to return class probabilities for all classifiers, including non-probabilistic classifiers like *svmRadial*. Classifier hyperparameters, i.e. model-intrinsic parameters that need to be adjusted to the studied data prior to modelling, were tuned for each classifier using a random search: 25 randomly chosen points in the hyperparameter space are evaluated and the point with the best performance metric (we chose the AUC in this study) is selected. The boundaries of the hyperparameter space are given in *caret*.

**Table 2.** Classifier characteristics.

| Classifier | *caret*[4] label | *R* package | Requires dummy coding | Tuned hyper-parameters |
|---|---|---|---|---|
| Elastic net logistic regression | *glmnet* | *glmnet*[25] | Yes | α, λ |
| Random forest | *rf* | *randomForest*[26] | No | *mtry* |
| Single-hidden-layer neural network | *nnet* | *nnet*[27] | No | *size, decay* |
| Support vector machine with radial basis function (RBF) kernel | *svmRadial* | *kernlab*[28] | Yes | σ, C |
| LogitBoost | *LogitBoost* | caTools[29] | Yes | *nlter* |
| Decision tree | *rpart* | *rpart*[30] | No | *cp* |

*Experimental Design*

For each classifier, test set (or *out-of-sample*) performance metrics (AUC, Brier score, accuracy, and Cohen's kappa) were estimated for each of the 12 datasets. The performance metric estimator was the average performance metric computed from the outer test folds in a nested and stratified 5-fold cross-validation (CV). The experiment was repeated 100 times. The 100 times repeated nested cross-validation yields a better estimate of the true test set performance by randomly simulating many scenarios with varying training and test set compositions.

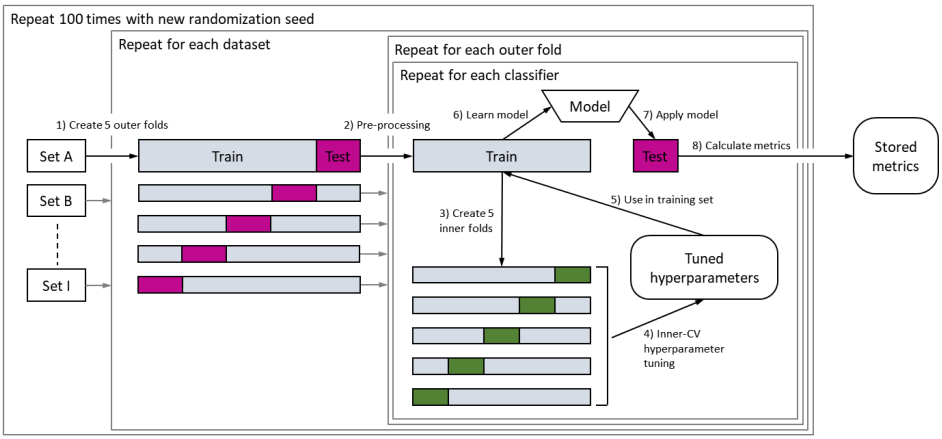



**Figure 1.** Experimental design: each dataset is split into 5 stratified outer folds (step 1). For each of the folds, the data is pre-processed (imputation, dummy coding, deleting zero variance features, rescaling) (step 2). The hyperparameters are tuned in the training set via a 5-fold inner CV (steps 3-5). Based on the selected hyperparameters, a model is learned on the training set (step 6) and applied on the test set (step 7). Performance metrics are calculated on the test set (step 8) and stored for all outer folds. This process is repeated 100 times for each classifier. Randomization seeds are stable across classifiers within a repetition to allow pairwise comparison.

The experimental design is depicted in figure 1: Each dataset was split into 5 random subsamples stratified for outcome classes (step 1 in figure 1), each of them acting once as a test set and 4 times as a part of a training set. The number of inner and outer folds was set to 5 following standard practice[24(p242)]. Data pre-processing is done per pair of training and test sets (step 2; see details in section *Datasets*). The models were trained on the training set (step 6) and applied on the test set (step 7) to compute the performance metrics for the test set (step 8) resulting in 5 estimates per performance metric (i.e. 1 per outer fold). During the training in each outer fold, the best tuning parameters were selected from the random search (see section *Classifiers*) according to the maximum AUC of an inner 5-fold CV. In the inner CV, the training set was again split into 5 subsamples and models with different tuning parameters were compared (steps 3-5). The nested 5-fold CV was repeated 100 times with different randomization seeds which are used, e.g., for generating the outer folds in step 1. Note that the performance metrics computed on the outer test folds of any two classifiers can be analysed by pairwise comparison because the classifiers were trained (step 6) and tested (step 7) on the same training and test sets for a specific dataset within each of the 100 repetitions.

The mean AUC, Brier score, accuracy, and Cohen's kappa were computed from the 5 estimates of the 5 folds in the outer CV. Calibration intercept and slope were computed from a linear regression of outcomes and predicted outcome probabilities for each of the 5 outer

folds. To attain aggregated calibration metrics over the 5 outer folds of the CV, the mean absolute differences from 0 and 1 were computed for the calibration intercept and slope, respectively. Classifier rankings were computed per dataset and repetition by ordering the classifiers' CV-mean AUC (i.e. the average AUC for 5 test sets) in descending order and then assigning the ranks from 1 to 6. Using CV-mean AUCs and CV-mean AUC *ranks*, we answer research questions 1 & 2. We chose AUC for the analysis following Steyerberg et al. (2010)[31]. They emphasize the importance of discrimination and calibration metrics when assessing prediction models. For the simplicity, we restricted the extended analysis to discrimination (AUC) but also report results for calibration and other metrics in appendix A.

To address the question of pre-selection (research question 3), we assess the advantage of choosing a classifier based on performance metrics from similar datasets, which we call *pre-selection* below. To estimate the benefit of our classifier pre-selection for a new dataset and to compare it to alternative strategies, the results of the experiment above were used as input for a simulation. For each outer fold of the 1200 5-fold CVs (12 datasets * 100 repetitions * 5 folds = 6000 folds), 3 classifier selections were made and tested on the test set that belongs to the specific outer fold:

- pre-selecting the classifier according to the average AUC *rank* in all other datasets (excluding all folds from the current dataset),
- selecting the classifier that performed best in the inner CV on the training set,
- randomly selecting a classifier.

Pre-selecting the classifier for one dataset that had the best average AUC *rank* in the other datasets simulates the scenario in which an investigator bases their classifier choice on empirical evidence as is reported in this manuscript. Randomly selecting a classifier represents the case where an investigator chooses a classifier without any prior knowledge about the dataset that (s)he is about to analyze. Selecting the tuned classifier with best inner CV performance corresponds to evaluating multiple classifiers on the training dataset and thus including dataset-specific information in the classifier selection. The performance metrics are averaged over all 500 outer folds (5 folds * 100 repetitions) for each of the 12 datasets.
The documented *R* code used for the analysis is available online[1].

## Results

Running 1 nested 5-fold cross-validation and computing the metrics on 1 dataset for all 6 classifiers allows 1 comparison of classifiers. This was applied on 12 different datasets, with each run repeated 100 times for a total of 1200 comparisons. The total computation time was approximately 6 days on an Intel Core i5-6200U CPU (or 15 seconds per classifier per dataset per outer fold, on average).

The results are presented and discussed threefold:

1. results aggregated over all datasets and repetitions to determine the presence of a superior classifier,
2. separate results for each dataset but aggregated over repetitions to determine dataset dependency,
3. a simulation of classifier selection methods in new datasets to estimate the relative effect of classifier pre-selection.

The detailed analysis is restricted to the classifiers' discriminative performance according to the AUC. Results for the remaining metrics (Brier score, calibration intercept/slope, accuracy, and Cohen's kappa) are reported in appendix A.

### *Results aggregated over all datasets*

Figure 2 shows the distribution of classifier rankings based on the average AUC (12 datasets * 100 repetitions = 1200 data points per classifier). Figure 3 depicts pairwise comparisons for each classifier pair (1200 comparisons per pair). The numbers in the plot indicate how often classifier A (y-axis) achieved an AUC greater than classifier B (x-axis). Coloring indicates whether the increased AUCs of classifier A are statistically significant (violet) or not (light violet). Untested pairs are colored grey. The significance cutoff was set to the 0.05-level (one-sided Wilcoxon signed-rank test, Holm-Bonferroni correction for 15 tests).

*rf* and *glmnet* showed the best median AUC *rank*, followed by *nnet*, *svmRadial*, *LogitBoost*, and *rpart* (figure 2). At the low end of the ranking, *rpart* showed poor discriminative performance. Manual inspection of the *rpart* models showed that *rpart* frequently returns empty decision trees for particular sets (for 34%, 67%, 35%, 58% of all outer folds for *sets D*, *F*, *K*, *L*, respectively). In pairwise comparisons, *rf* and *glmnet* significantly outperformed all other classifiers (figure 3). *rf* exhibited a small but statistically insignificant better AUC *rank* than *glmnet*.

The results in figures 2 and 3 indicate the existence of a significant classifier ranking for these datasets. However, the considerable spread per classifier in figure 2 and the low pairwise comparison percentages (between 57% and 88% in figure 3) also suggest a yet unobserved dependency for classifier performance. To this end, the relationship between datasets and varying classifier performance is investigated.

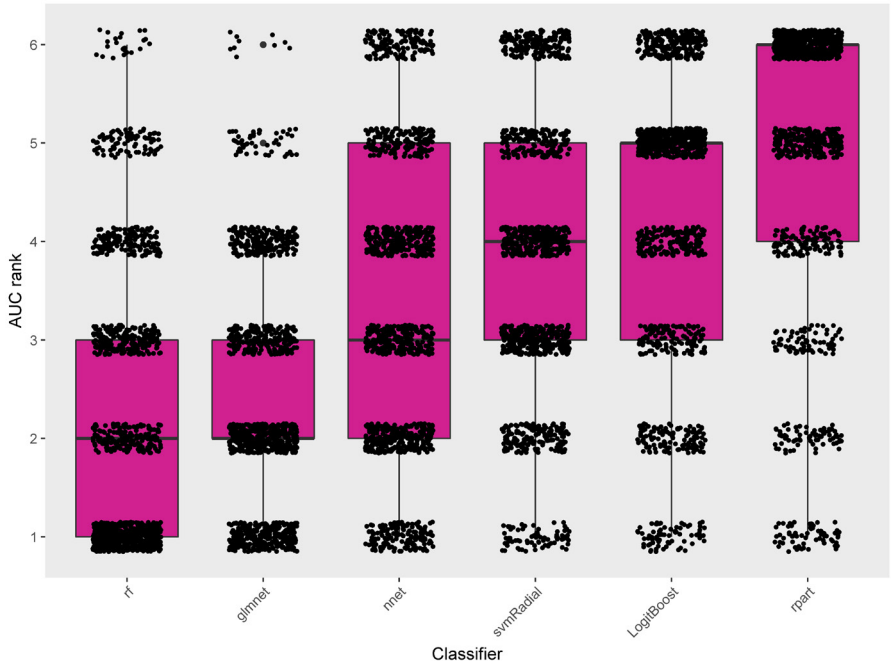



**Figure 2.** Box- and scatterplot of the AUC *rank* (lower being better) per outer 5-fold CV aggregated over all datasets and repetitions (12 datasets * 100 repetitions = 1200 data points per classifier).

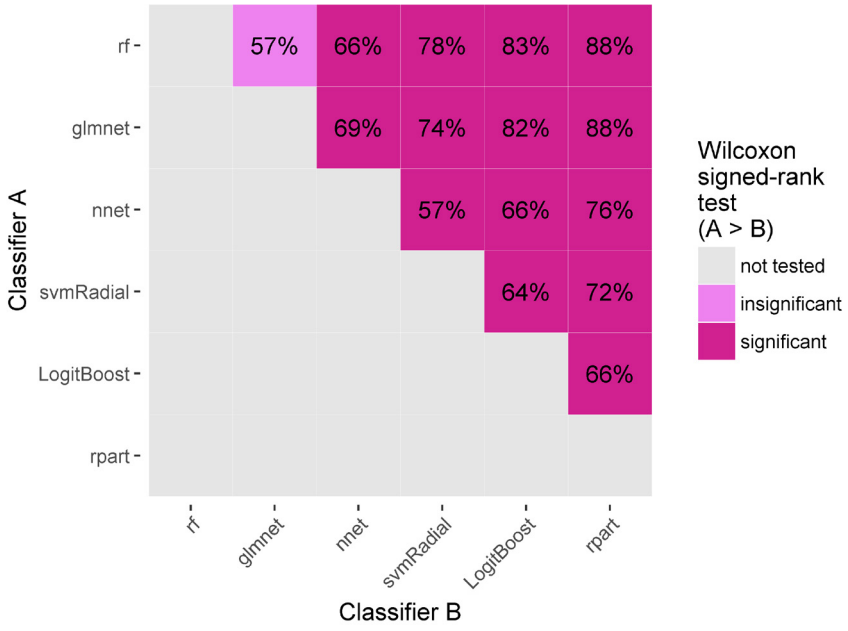



**Figure 3.** Pairwise comparisons of each classifier pair (12 datasets * 100 repetitions = 1200 comparisons per pair). The numbers in the plot indicate how often classifier A (y-axis) achieved an AUC greater than classifier B (x-axis). The color indicates whether the increased AUCs by classifier A are statistically significant (violet), insignificant (light violet), or have not been tested (grey). The significance cutoff was set to the 0.05-level (one-sided Wilcoxon signed-rank test, Holm-Bonferroni correction for 15 tests).

*Results separate for each dataset*

Figure 4 shows the average AUC for each pair of classifier and dataset (100 repetitions = 100 data points per pair). Figure 5 depicts the average *rank* derived from the AUC (100 data points per pair).

*rf* and *glmnet* generally yielded higher AUC values and AUC *ranks* per dataset (figures 4 & 5). However, this observation is not consistent over all datasets: e.g., *nnet* outperforms *rf* in *sets G*, *J*, and *K,* and *svmRadial* outperformed *glmnet* in *sets A* and *C*.

The results in the figures 4 and 5 indicate that dataset-specific properties impact the discriminative performance of classifiers. These results challenge our proposition that one can pre-select classifiers for predictive modelling in (chemo)radiotherapy based on representative datasets from the same field.

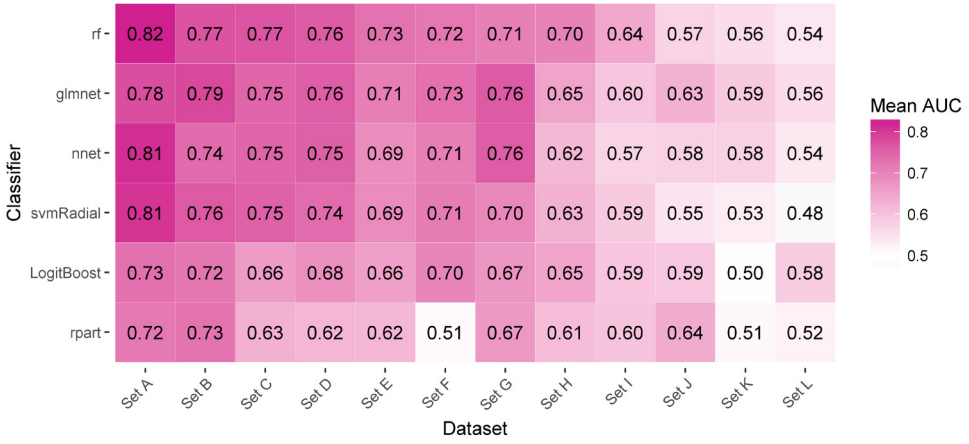



**Figure 4.** The mean AUC for each pair of classifier and dataset (100 repetitions = 100 data points per pair).

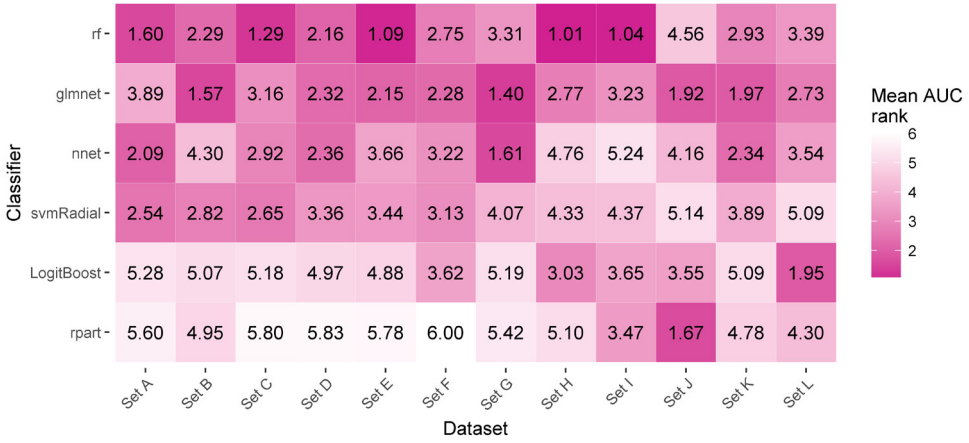



**Figure 5.** The mean *rank* derived from the AUC (100 repetitions = 100 data points per pair).

*Effects of empirical classifier pre-selection on discriminative performance*

Table 3 lists, for each dataset, the name and average AUCs, i.e. averaged over all 100 repetitions, for random classifier selection, classifier pre-selection, and set-specific classifier selection.

The pre-selection procedure always results in *rf* or *glmnet*. The mean benefit of empirically pre-selecting a classifier is small: the AUC improvement ranges between -0.01 and 0.07 with a mean of 0.02. In a pairwise comparison over all datasets ($p < 0.05$, one-sided Wilcoxon signed-rank test), the AUC values by pre-selection were significantly larger than the AUC values by random selection. The AUC *rank* improves by 0.52 on average. Including dataset-specific information by inner CV yields a mean AUC improvement of 0.02 and improves the *rank*, on average, by 0.65. In a pairwise comparison of set-specific and random classifier selection over all datasets ($p < 0.05$, one-sided Wilcoxon signed-rank test), the AUC increase was also statistically significant.

Given this simulation, the expected benefit of pre-selecting a classifier for a new dataset based on results from (chemo)radiotherapy-specific numerical studies is limited with an average increase in AUC of 0.02.



**Table 3.** For each dataset, the AUC *rank* averaged over all repetitions when (a) randomly selecting a classifier (Random classifier), (b) pre-selecting the classifier with the average best AUC *rank* in all other datasets, i.e. without any information about the current dataset (Pre-selected classifier), (c) selecting the classifier that yielded the highest AUC in the inner CV (Set-specific classifier). Improvements in average AUC and average AUC *rank* compared to (a) are reported. The average AUC improvements by pre-selection and set-specific selection were tested for statistical significance ($p < 0.05$, one-sided Wilcoxon signed-rank test) and found to be statistically significant (*). No other statistical tests besides the two aforementioned tests were conducted.

| | **Random classifier** | | **Pre-selected classifier** | | | **Set-specific classifier** | | |
|---|---|---|---|---|---|---|---|---|
| **Dataset** | **Rank** | **Name** | **Rank** | | **AUC** | **Rank** | | **AUC** |
| | **Mean** | | **Mean** | **Increase** | **Increase** | **Mean** | **Increase** | **Increase** |
| Set A | 3.43 | *glmnet* | 3.64 | -0.21 | 0.00 | 3.10 | 0.33 | 0.02 |
| Set B | 3.44 | *rf* | 2.92 | 0.52 | 0.02 | 3.31 | 0.13 | 0.00 |
| Set C | 3.49 | *rf* | 1.94 | 1.55 | 0.05 | 2.78 | 0.71 | 0.03 |
| Set D | 3.59 | *rf* | 2.60 | 0.99 | 0.05 | 3.31 | 0.28 | 0.02 |
| Set E | 3.53 | *rf* | 1.89 | 1.63 | 0.05 | 2.58 | 0.94 | 0.03 |
| Set F | 3.57 | *rf* | 2.99 | 0.58 | 0.04 | 3.52 | 0.05 | 0.01 |
| Set G | 3.43 | *rf* | 3.81 | -0.39 | 0.00 | 1.70 | 1.73 | 0.05 |
| Set H | 3.65 | *rf* | 1.59 | 2.06 | 0.07 | 1.71 | 1.93 | 0.06 |
| Set I | 3.49 | *glmnet* | 3.50 | 0.00 | 0.00 | 2.08 | 1.42 | 0.03 |
| Set J | 3.52 | *rf* | 4.18 | -0.67 | -0.01 | 3.41 | 0.11 | 0.01 |
| Set K | 3.59 | *rf* | 3.33 | 0.26 | 0.02 | 3.20 | 0.39 | 0.02 |
| Set L | 3.44 | *rf* | 3.50 | -0.06 | 0.00 | 3.66 | -0.22 | -0.01 |
| **Mean** | **3.51** | | **2.99** | **0.52** | **0.02*** | **2.86** | **0.65** | **0.02*** |

## Discussion

Our results suggest that there is indeed an overall ranking of classifiers in (chemo)radiotherapy datasets, with *rf* and *glmnet* leading the ranking. However, we also observe that the performance of a classifier depends on the specific dataset. Pre-selecting classifiers based on evidence from related datasets would, on average, provide a benefit for investigators because it increases discriminative performance. An increase in average discriminative performance is desirable in that an investigator would be less likely to discard their data because of a perceived absence of predictive or prognostic value. The estimated 0.02 mean AUC improvement might appear small but it comes 'for free' with classifier selection based on empirical evidence from multiple radiotherapy datasets. Furthermore, the 0.02 AUC improvement is relative to random classifier selection. If an investigator had initially chosen *rpart,* which is the overall worst performing classifier in our study, switching to the preselected classifier would result in an average AUC increase of 0.07. Switching from LogitBoost, which is the second worst performing classifier in our study, to the preselected classifier would result in an average AUC increase of 0.04. The results in table 3 show that classifier pre-selection and set-specific classifier selection, on average, yield the same AUC increase. We think that the usefulness of set-specific classifier selection is dependent on the size of the training set: classifier pre-selection is preferable for small datasets, set-specific classifier selection is better for larger datasets. Classifier pre-selection represents choosing classifiers using evidence from a large collection of similar datasets from the general radiotherapy outcome domain. Set-specific classifier selection represents choosing classifiers based on the training set, which is a considerably smaller evidence base but comes from the patient group under investigation. If the training dataset is too small, selecting classifiers based on results from other datasets might be less-error prone. On the contrary, if an investigator has collected a large dataset, they have the option to conduct set-specific classifier selection (with all 6 classifiers) for their training data using our documented *R* code[1].

In table 3, one can observe that the pre-selected classifier is mostly *rf* and sometimes *glmnet*. To understand this behaviour, consider dataset *A*: *glmnet* was pre-selected for *set A* by selecting the classifier with the best average AUC *rank* in all other sets (excluding *set A*). Note that, for all 12 datasets together, the average AUC *rank* for *rf* is only slightly better than for *glmnet* (2.29 for *rf* and 2.45 for *glmnet*; the average of the rows in figure 5). Since *glmnet* performs badly while *rf* performs best in *set A*, excluding this information leads to a better average AUC *rank* for *glmnet* and a worse average AUC *rank* for *rf* in the remaining 11 datasets. As a consequence, *glmnet* becomes the pre-selected classifier for this dataset. A similar behaviour is observed for *set I* but not in *sets C*, *D*, *E*, *H*, where *glmnet* also performs worse than *rf* but the difference between both classifiers is smaller and does not induce a switch in the pre-selected classifier.

The result that classifier pre-selection is as good as set-specific selection in the studied datasets does *not* imply that one *cannot* determine a better classifier for a new dataset. Our implementation of set-specific classifier selection only evaluates the performance of various classifiers but does not directly take into account properties of the dataset itself. For example, if an investigator collected a dataset in which the outcome has a quadratic dependency on a feature, *glmnet* would not be able to capture this relation (since it models only linear effects) but *rf* would. However, pre-selecting a classifier based on results from other (chemo) radiotherapy datasets works well on average. Furthermore, including set-specific classifier selection complicates the modelling process and therefore might not be desirable.

In this study, we collected 12 datasets for different treatment sites, i.e. (non-) small cell lung cancer, head and neck cancer, meningioma with different outcomes, i.e. survival, pneumonitis, esophagitis, odynophagia, regional control. However, this collection is certainly not a complete representation of treatment outcome datasets analyzed in the field of radiotherapy. Furthermore, we only studied one implementation of classifiers while classifier performance may vary between implementations. Past studies, however, indicate that classifier implementations in *R* interfaced with *caret* are competitive[5]. Given the apparent lack of comparative classifier studies in radiotherapy, our intention has been to provide numerical evidence for classifier selection to investigators even though our analysis is not exhaustive.

We intentionally limited the analysis to classifier selection while ignoring factors such as the investigator's experience, usage in literature, hypothetical feature dependencies, and model interpretability. This restriction imitates the current trend to delegate modelling decisions to machine learning algorithms and/or non-domain experts. Nonetheless, we feel the need to emphasize that including these factors has merit. Furthermore, expertise on a specific classifier could warrant its selection: Lavesson and Davidsson (2006)[32] observed in a study on 8 datasets from different research domains that the impact of hyperparameter tuning exceeds that of classifier selection. Therefore, the investigator could tune a classifier for better performance by also tuning the hyperparameters outside the subset of hyperparameters tuneable inside *caret*. Even in those cases, however, we suggest comparing these results to simpler implementations of *rf* and *glmnet* as these classifiers on average have the best discriminative performance according to this study. Finally, for the clinical implementation of classifiers, model interpretability is arguably a major requirement[33]: this view is also convincingly motivated by Caruana et al.[34]. Fortunately, our study shows that *glmnet*, which is an intuitive classifier, is also one of the best performing classifiers.



## Conclusion

We have modelled treatment outcomes in 12 datasets using 6 different classifier implementations in the popular open-source software *R* interfaced with the package *caret*. Our results provide evidence that the easily interpretable elastic net logistic regression and the complex random forest classifiers generally yield higher discriminative performance in (chemo)radiotherapy outcome and toxicity prediction than the other classifiers. Thus, one of these two classifiers should be the first choice for investigators to build classification models or to compare one's own modelling results. Our results also show that an informed pre-selection of classifiers based on existing datasets improves discrimination over random selection.

## Disclosure of Conflicts of Interest

Andre Dekker, Johan van Soest, Tim Lustberg are founders and shareholders of Medical Data Works B.V., which provides consulting on medical data collection and analysis projects. Cary Oberije is CEO of ptTheragnostic B.V. Philippe Lambin is member of the advisory board of ptTheragnostic B.V.


## Acknowledgements

Authors acknowledge financial support from ERC advanced grant (ERC-ADG-2015, n° 694812 - Hypoximmuno) and the QuIC-ConCePT project, which is partly funded by EFPI A companies and the Innovative Medicine Initiative Joint Undertaking (IMI JU) under Grant Agreement No. 115151. This research is also supported by the Dutch technology Foundation STW (grant n° 10696 DuCAT & n° P14-19 Radiomics STRaTegy), which is the applied science division of NWO, and the Technology Programme of the Ministry of Economic Affairs. Authors also acknowledge financial support from the EU 7th framework program (ARTFORCE - n° 257144, REQUITE - n° 601826), SME Phase 2 (RAIL - n°673780), EUROSTARS (SeDI, CloudAtlas, DART, DECIDE), the European Program H2020 (BD2Decide - PHC30-689715, ImmunoSABR - n° 733008, PREDICT - ITN - n° 766276, CLEARLY - TRANSCAN-FP-045), Interreg V-A Euregio Meuse-Rhine ("Euradiomics"), Kankeronderzoekfonds Limburg from the Health Foundation Limburg, Alpe d'HuZes-KWF (DESIGN), the Zuyderland-MAASTRO grant and the Dutch Cancer Society, KWF- TraIT2HealthRI, Province Limburg-LIME-Personal Health Train, NFU-Data4LifeSciences, Varian Medical Systems-SAGE & ROO.

## References


1. Deist TM, Dankers FJWM, Valdes G, *et al.* Code for: Machine learning algorithms for outcome prediction in (chemo)radiotherapy: an empirical comparison of classifiers. https://github.com/timodeist/classifier_selection_code.
2. Lambin P, van Stiphout RGPM, Starmans MHW, *et al.* Predicting outcomes in radiation oncology--multifactorial decision support systems. *Nat Rev Clin Oncol*. 2013;**10**(1):27-40. doi:10.1038/nrclinonc.2012.196
3. Lambin P, Roelofs E, Reymen B, *et al.* 'Rapid Learning health care in oncology' – An approach towards decision support systems enabling customised radiotherapy'. *Radiother Oncol*. 2013;**109**(1):159-164. doi:10.1016/j.radonc.2013.07.007
4. Kuhn M, Wing J, Weston S, *et al. Caret: Classification and Regression Training*.; 2016. https://CRAN.R-project.org/package=caret.
5. Fernández-Delgado M, Cernadas E, Barro S, Amorim D. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? *J Mach Learn Res*. 2014;**15**:3133-3181.
6. Wainer J. Comparison of 14 different families of classification algorithms on 115 binary datasets. *ArXiv160600930 Cs*. June 2016. http://arxiv.org/abs/1606.00930. Accessed April 8, 2017.
7. Olson RS, Cava WL, Mustahsan Z, Varik A, Moore JH. Data-driven advice for applying machine learning to bioinformatics problems. In: *Biocomputing 2018*. WORLD SCIENTIFIC; 2017:192-203. doi:10.1142/9789813235533_0018
8. Parmar C, Grossmann P, Bussink J, Lambin P, Aerts HJWL. Machine Learning methods for Quantitative Radiomic Biomarkers. *Sci Rep*. 2015;**5**:13087. doi:10.1038/srep13087
9. Belderbos J, Heemsbergen W, Hoogeman M, Pengel K, Rossi M, Lebesque J. Acute esophageal toxicity in non-small cell lung cancer patients after high dose conformal radiotherapy. *Radiother Oncol*. 2005;**75**(2):157-164. doi:10.1016/j.radonc.2005.03.021
10. Bots WTC, van den Bosch S, Zwijnenburg EM, *et al.* Reirradiation of head and neck cancer: Long-term disease control and toxicity. *Head Neck*. 2017;**39**(6):1122-1130. doi:10.1002/hed.24733
11. Carvalho S, Troost EGC, Bons J, Menheere P, Lambin P, Oberije C. Prognostic value of blood-biomarkers related to hypoxia, inflammation, immune response and tumour load in non-small cell lung cancer – A survival model with external validation. *Radiother Oncol*. 2016;**119**(3):487-494. doi:10.1016/j.radonc.2016.04.024
12. Carvalho S, Troost E, Bons J, Menheere P, Lambin P, Oberije C. Data from: Prognostic value of blood-biomarkers related to hypoxia, inflammation, immune response and tumour load in non-small cell lung cancer – a survival model with external validation. http://doi.org/10.17195/candat.2016.04.1. Published 2016.
13. Janssens GO, Rademakers SE, Terhaard CH, *et al.* Accelerated Radiotherapy With Carbogen and Nicotinamide for Laryngeal Cancer: Results of a Phase III Randomized Trial. *J Clin Oncol*. 2012;**30**(15):1777-1783. doi:10.1200/JCO.2011.35.9315
14. Jochems A, Deist TM, El Naqa I, *et al.* Developing and Validating a Survival Prediction Model for NSCLC Patients Through Distributed Learning Across 3 Countries. *Int J Radiat Oncol*. 2017;**99**(2):344-352. doi:10.1016/j.ijrobp.2017.04.021
15. Kwint M, Uyterlinde W, Nijkamp J, *et al.* Acute Esophagus Toxicity in Lung Cancer Patients After Intensity Modulated Radiation Therapy and Concurrent Chemotherapy. *Int J Radiat Oncol • Biol • Phys*. 2012;**84**(2):e223-e228. doi:10.1016/j.ijrobp.2012.03.027
16. Egelmeer AGTM, Velazquez ER, Jong JMA de, *et al.* Development and validation of a nomogram for prediction of survival and local control in laryngeal carcinoma patients treated with

radiotherapy alone: A cohort study based on 994 patients. *Radiother Oncol*. 2011;**100**(1):108-115. doi:10.1016/j.radonc.2011.06.023
17. Lustberg T, Bailey M, Thwaites DI, *et al.* Implementation of a rapid learning platform: Predicting 2-year survival in laryngeal carcinoma patients in a clinical setting. *Oncotarget*. 2016;**7**(24):37288-37296. doi:10.18632/oncotarget.8755
18. Oberije C, De Ruysscher D, Houben R, *et al.* A Validated Prediction Model for Overall Survival From Stage III Non-Small Cell Lung Cancer: Toward Survival Prediction for Individual Patients. *Int J Radiat Oncol Biol Phys*. 2015;**92**(4):935-944. doi:10.1016/j.ijrobp.2015.02.048
19. Oberije C, De Ruysscher D, Houben R, *et al.* Data from: A validated prediction model for overall survival from Stage III Non Small Cell Lung Cancer: towards survival prediction for individual patients. 2015. https://www.cancerdata.org/id/10.5072/candat.2015.02.
20. Olling K, Nyeng DW, Wee L. Predicting acute odynophagia during lung cancer radiotherapy using observations derived from patient-centred nursing care. *Tech Innov Patient Support Radiat Oncol*. 2018;**5**:16-20. doi:10.1016/j.tipsro.2018.01.002
21. Wijsman R, Dankers F, Troost EGC, *et al.* Multivariable normal-tissue complication modeling of acute esophageal toxicity in advanced stage non-small cell lung cancer patients treated with intensity-modulated (chemo-)radiotherapy. *Radiother Oncol*. 2015;**117**(1):49-54. doi:10.1016/j.radonc.2015.08.010
22. Wijsman R, Dankers F, Troost EGC, *et al.* Inclusion of incidental radiation dose to the cardiac atria and ventricles does not improve the prediction of radiation pneumonitis in advanced stage non-small cell lung cancer patients treated with intensity-modulated radiation therapy. *Int J Radiat Oncol*. doi:10.1016/j.ijrobp.2017.04.011
23. James G, Witten D, Hastie T, Tibshirani R. *An Introduction to Statistical Learning: With Applications in R*. New York: Springer-Verlag; 2013. //www.springer.com/gp/book/9781461471370. Accessed March 4, 2018.
24. Hastie T, Tibshirani R, Friedman J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. 2nd ed. New York: Springer-Verlag; 2009. //www.springer.com/gp/book/9780387848570. Accessed March 4, 2018.
25. Friedman J, Hastie T, Tibshirani R. Regularization Paths for Generalized Linear Models via Coordinate Descent. *J Stat Softw*. 2010;**33**(1):1–22.
26. Liaw A, Wiener M. Classification and Regression by randomForest. *R News*. 2002;**2**(3):18-22.
27. Venables WN, Ripley BD. *Modern Applied Statistics with S*. Fourth. New York: Springer; 2002. http://www.stats.ox.ac.uk/pub/MASS4.
28. Karatzoglou A, Smola A, Hornik K, Zeileis A. kernlab – An S4 Package for Kernel Methods in R. *J Stat Softw*. 2004;**11**(9):1–20.
29. Tuszynski J. *CaTools: Tools: Moving Window Statistics, GIF, Base64, ROC AUC, Etc.*; 2014. https://CRAN.R-project.org/package=caTools.
30. Therneau T, Atkinson B, Ripley B. *Rpart: Recursive Partitioning and Regression Trees.*; 2017. https://CRAN.R-project.org/package=rpart.
31. Steyerberg EW, Vickers AJ, Cook NR, *et al.* Assessing the performance of prediction models: a framework for traditional and novel measures. *Epidemiol Camb Mass*. 2010;**21**(1):128-138. doi:10.1097/EDE.0b013e3181c30fb2
32. Lavesson N, Davidsson P. Quantifying the Impact of Learning Algorithm Parameter Tuning. In: *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*. Boston, Massachusetts: AAAI Press; 2006:395–400. http://dl.acm.org/citation.cfm?id=1597538.1597602. Accessed April 9, 2017.

33. Valdes G, Luna JM, Eaton E, Ii CBS, Ungar LH, Solberg TD. MediBoost: a Patient Stratification Tool for Interpretable Decision Making in the Era of Precision Medicine. *Sci Rep*. 2016;**6**:37854. doi:10.1038/srep37854
34. Caruana R, Lou Y, Gehrke J, Koch P, Sturm M, Elhadad N. Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '15. New York, NY, USA: ACM; 2015:1721–1730. doi:10.1145/2783258.2788613

## Appendix A

Table A1 lists performance metrics per classifier. These values are averaged over all repetitions and datasets (100 repetitions * 12 datasets = 1200 data points each). Accuracy and Cohen's kappa were computed at the 0.5-cutoff. Calibration fails in some outer folds for every classifier resulting in either large or undefined values for intercept and/or slope. This failure occurs frequently with *nnet* and *rpart*. Undefined (NaN) values are excluded when calculating the median.

**Table A1.** Median performance metrics per classifier aggregated over repetitions and datasets (1200 data points each). Undefined (NaN) values are excluded when calculating the median.

| Classifier | AUC | Brier score | Accuracy | Cohen's kappa | Calibration intercept error | Calibration slope error |
|---|---|---|---|---|---|---|
| ***rf*** | 0.71 | 0.19 | 0.70 | 0.14 | 0.12 | 0.38 |
| ***glmnet*** | 0.71 | 0.20 | 0.70 | 0.14 | 0.26 | 0.66 |
| ***nnet*** | 0.69 | 0.22 | 0.67 | 0.11 | 0.31 | 0.87 |
| ***svmRadial*** | 0.69 | 0.19 | 0.70 | 0.06 | 0.32 | 0.82 |
| ***LogitBoost*** | 0.66 | 0.24 | 0.66 | 0.18 | 0.24 | 0.60 |
| ***rpart*** | 0.62 | 0.23 | 0.67 | 0.17 | 0.22 | 0.55 |

# Chapter 5

## Simulation assisted machine learning

Timo M. Deist, Andrew Patti, Zhaoqi Wang, David Krane, Taylor Sorenson, David Craft

*Submitted*

## Abstract

**Motivation:** In a predictive modeling setting, if sufficient details of the system behavior are known, one can build and use a simulation for making predictions. When sufficient system details are not known, one typically turns to machine learning, which builds a black-box model of the system using a large dataset of input sample features and outputs.

We consider a setting which is between these two extremes: some details of the system mechanics are known but not enough for creating simulations that can be used to make high quality predictions. In this context we propose using approximate simulations to build a kernel for use in kernelized machine learning methods, such as support vector machines. The results of multiple simulations (under various uncertainty scenarios) are used to compute similarity measures between every pair of samples: sample pairs are given a high similarity score if they behave similarly under a wide range of simulation parameters. These similarity values, rather than the original high dimensional feature data, are used to build the kernel.

**Results:** We demonstrate and explore the simulation based kernel (SimKern) concept using four synthetic complex systems--three biologically inspired models and one network flow optimization model. We show that, when the number of training samples is small compared to the number of features, the SimKern approach dominates over no-prior-knowledge methods. This approach should be applicable in all disciplines where predictive models are sought and informative yet approximate simulations are available.

**Availability:** The Python SimKern software, the demonstration models (in MATLAB, R), and the datasets are available at **https://github.com/davidcraft/SimKern**.

## Introduction and motivation

There are two general approaches to computationally predicting the behavior of complex systems, simulation and machine learning (ML). Simulation is the preferred method if the dynamics of the system being studied are known in sufficient detail that one can simulate its behavior with high fidelity and map the system behavior to the output to be predicted. ML is valuable when the system defies accurate simulation but enough data exists to train a general black-box machine learner, which could be anything from a linear regression or classification model to a neural network. In this work, we propose a technique to combine simulation and ML in order to leverage the best aspects of both and produce a system that is superior to either technique alone.

Our motivation is personalized medicine: how do we assign the right drug or drug combination to cancer patients? Across cultures and history, physicians prescribe medicines and interventions based on how the patient is predicted to respond. Currently these choices are made based on established patient-classification protocols, physician judgment, clinical trial eligibility, and occasionally limited genomic profiling of the patient. All of these approaches, in one way or another, attempt to partition patients into groups based on some notion of similarity.

Genomics is especially relevant for computing the similarity between two cancer patients since cancer is associated with alterations to the DNA, which in turn causes the dysregulation of cellular behavior[1]. Bioinformatic analysis has revealed that there is heterogeneity both within a patient tumor and across tumors; no two tumors are the same genomically[2,3]. Although in a small fraction of cases specific genetic conditions are used to guide therapy choices, for example breast (commonly amplified gene: HER2), melanoma (BRAF mutation), lung (EML4-ALK fusion), and head-and-neck (HPV status for radiation dose de-escalation[4]), there remains a large variability in patient responses to these and other treatments, likely due to the fact that patients will usually have tens or hundreds of mutations and gene copy number variations, chromosomal structural rearrangements, not to mention a distinct germline genetic state[5], human leukocyte antigen type[6], tumor epigenetic DNA modifications, microbiome, and comorbidity set. Even amidst this heterogeneity, the notion of patient similarity—although currently not deeply understood due to the complexities of cancer biology—is appealing both conceptually and for its value in the ML setting.

Simulating a drug is a task that far exceeds our current scientific capacity: it enters the patient, either intravenously or orally, and winds its way to the cancer cells, where it either influences the cancer cell via receptors on the cell membrane or penetrates into the cell and affects signaling pathways, cell metabolism, DNA repair, apoptosis, or some combination of these and other modules. Nevertheless, a vast amount of knowledge of cellular processes, residing in molecular biology textbooks and millions of scientific papers, has been accrued over the past century and it seems worthwhile to attempt to use that information, if unclear how. Most machine learning research efforts in the personalized medicine realm take a pure data approach. Given the complexity of patient biology and cancer, this approach will require vast amounts of high quality patient data that is suitably standardized for algorithmic processing

With this drug sensitivity prediction problem as our backdrop, we develop a method to combine approximate simulations with ML and demonstrate using *in silico* experiments that a judicious combination can yield better predictions than either technique alone. The basic idea is a division of labor: coarse and approximate simulations are used to compute similarity measures, and these similarity measures are then used by the ML algorithm to build a predictive model, called SimKern ML (Figure 1).

At this point in time, although vast details of cellular biology are known, we are not in a position to simulate with any fidelity complete cellular or *in vivo* cancerous processes. However, herein we present demonstrations that one could combine simulation results into machine learning and improve the overall predictive capability, a technique which may play a role in future drug recommendation systems.

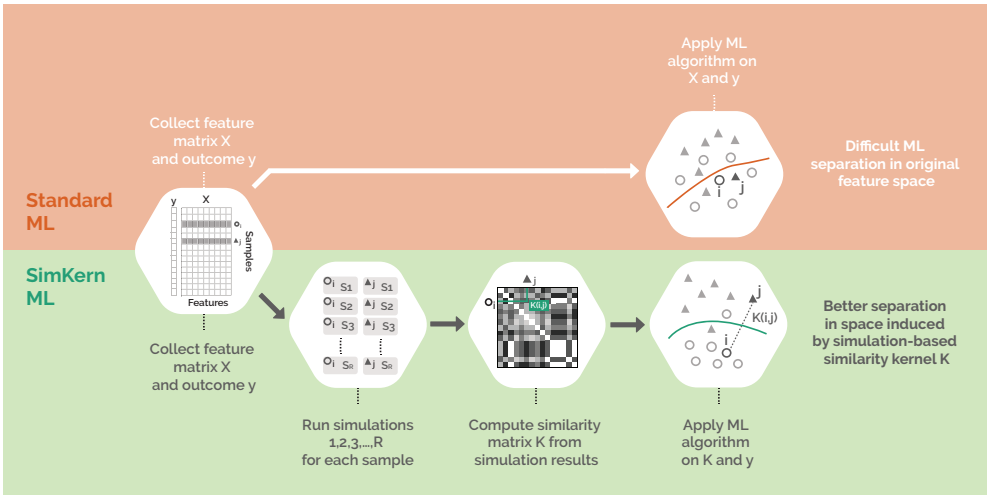



**Figure 1.** Workflow comparison of Standard machine learning (ML) and SimKern ML. The feature matrix *X* and outcome data *y* are given (in this paper, we generate such "ground truth" datasets by simulating complex systems, a step which is not shown in this figure). Traditional feature-based ML is depicted in the upper orange part. SimKern, the simulation-based method, pre-processes the dataset by sending each sample through a number of approximate simulations. Each sample pair is given a similarity score based on how closely they behave under the various simulations (See Figure 2 and Section SimKern simulation – similarity matrix generation for more details). This information is stored in a kernel matrix *K* where *K(i,j)* measures the similarity between samples *i* and *j* Note that $K(i,i)=1$ and $0 \leq K(i,j) \leq 1$. Useful SimKern simulations yield a kernel *K* that improves the downstream machine learning performance.

## Materials and methods

Our method is centered on kernelized ML. Rather than feature vectors (a list of attributes for each sample), kernelized learning requires only a similarity score between pairs of samples. For training, one needs the outcome of each training sample and a measurement of the similarity between all pairs of training samples. For predicting the outcome of a new sample, one needs to provide the similarity of that sample to each training sample. It is well known in ML that good similarity measures, which come from expert domain knowledge, result in better ML performance[7]. We assume that we can formulate a simulation of each sample's behavior based on its known individual characteristics (i.e. features). We also assume that we do not know exactly how to simulate the systems, so rather than a single simulation we have a family (possibly parametrized by real numbers, and thus infinite) of plausible simulations. Two samples are given a high similarity score if they behave similarly across a wide range of simulations.

We begin with a brief description of the four models we use to demonstrate and analyze the performance of SimKern. By describing these models, the reader has in mind a more concrete context with which to frame the SimKern development.

*Brief model descriptions*

We investigate four models: radiation impact on cells, flowering time in plants, a Boolean cancer model, and a network flow optimization problem. Full details and model implementation notes are given in the Supplementary information.

For each model we begin by generating a dataset of $N$ samples, each sample $i$ is described by a feature vector $x_i$ of length $p$ and a response $y_i$ using the ground truth simulation (see Figure 1). This produces an $N \times p$ feature matrix $X$ and a response vector $y$ of length $N$. This ground truth simulation (referred to as SIM0 in the code repository) is not part of our kernelized learning method, but the datasets created are needed to demonstrate the simulation-based kernel ML method. This ground truth simulation step is further described in the Supplementary information. In an actual application of SimKern, this artificial data creation step would not be used.



The **radiation cancer cell death model** is a set of ordinary differential equations (ODEs) which represents a simplified view of the biochemical processes that happen after a cancer cell is hit by radiation. The core of the model involves the DNA damage response regulated by the phosphorylation of ATM and subsequent p53 tetramerization[8]. We have added cell cycle arrest terms, a DNA repair process, and apoptosis modules in order to capture the idea that cellular response to DNA damage involves the combined dynamics of these various processes. The model, which is depicted as a network graph, is displayed in Supplementary figure S3, and consists of 34 ODEs. The rate parameters were not tuned to realistic values (except for the ones from the original P53 core network, where we used the values provided by the authors[8]). Instead, values were manually chosen such that the family of samples created had representatives in each of the four output classes: apoptosis, repaired and cycling, mitotic catastrophe, and quiescence. A population of distinct cell types is formed by varying 33 of the ODE rate constants and the mutation status of six genes (ARF, BAX, SIAH, Reprimo, p53, and APAF1), for a feature vector length of 39. The SimKern simulation uses the same underlying model as the original ODE model with two key differences: 87 of the ODE parameters are marked as uncertain and given Gaussian probability distributions around their true values, and the simulation outputs the time dynamics of the ODEs rather than a classification.

The **flowering time model** is a set of six ODEs that simulate the gene regulatory network governing the flowering of the Arabidopsis plant[9], and yields a regression problem. 19 mutants are modeled and experimentally validated by the authors. We use those 19 mutational states as well as 34 additional perturbations on the rate parameters to create a varied ground truth sample set. The output of the model is the time to flowering which, following the authors, is set to the time at which the protein AP1 exceeds a particular threshold. For the SimKern model we assume the same model but with uncertainty about the rate parameters. The SimKern simulation output is the time dynamics of the six ODEs.

The **Boolean cancer model** is a discrete dynamical system of cancer cellular states[10]. Based on the steady state of the system, a sample is labelled as one of three categories: apoptotic, metastasizing, or other. There are no rate parameters since this is a Boolean model. We use the initial state vector (the on/off status of the 32 nodes in the network) as well as mutations of five of the genes (p53, AKT1, AKT2, NICD, and TGF$\beta$) to create a varied sample population with 37 features. In the SimKern simulation, we use a reduced version of the model provided in the original publication. It is unclear how to map the initial conditions from the full model to the initial conditions of the modularly-reduced model, so for all of the modules we randomly choose the mapping, which gives rise to the uncertainty for the SimKern simulations. The output from the SimKern model (i.e. the data used to form the similarity matrix) is the same classification as from the ground truth model.

The **network flow model** is an optimization problem rather than a simulation. It falls into a subclass of linear optimization models called network flows which are used in a wide range of applications including production scheduling and transportation logistics[11]. The network flow model takes arc costs as inputs, which are the costs of sending a unit of flow through a certain arc in the network. The model then simulates the optimal path of flow along arcs of a directed graph that minimizes the total arc cost along the path. The network is designed in layers and is such that the flow will pass through exactly one of the three arcs in the final layer, which gives us a classification problem (see Figure S4). Changes in arc costs, which represent the features in this model, can lead to changes in the routing of the optimal flow. For the ground truth dataset, we generate samples by varying 12 out of the 80 arc costs. We build two separate SimKern simulations: the better simulation perturbs 23 arc costs, including the 12 costs that were varied to make the ground truth dataset, resulting in a less noisy kernel. The worse simulation varies 21 additional arc costs resulting in a noisier kernel.

*SimKern simulation – similarity matrix generation*

Users must define a model (currently supported languages for the simulation modeling are MATLAB, Octave, and R) which simulates a sample. This simulation procedure, called SIM1 in the python codebase, is used to generate the sample similarity kernel matrix and would be the starting point in an actual application of SimKern. Figure 2 illustrates the SimKern simulation process control.

We assume that there are parameters in this simulation model that we are uncertain about. Let $\theta$ be a vector of these uncertain parameters. We assume we have a random variable description of each of these parameters, which can be very general. For example, a parameter could take the value of 0 or 1 if we have two ways of modeling a particular interaction.

Then, in the simulation, depending on how that random variable gets instantiated, the code uses one of the two parameter values. Alternatively, we might be uncertain about the value of a rate constant, in which case we could use a Gaussian random variable with a specified mean and standard deviation. We assume independence of the random variables $\theta$, but one could also assume a covariance structure.

Each sample $i = 1...\ N$ is characterized by a feature vector $x_i$, which constitutes sample-specific information that we use to perform the simulations; $x_i$ could be for example a genomic description of patient *i*. For $r = 1...\ R$, where $R$ is the number of trials to run, we instantiate a parameter vector, $\theta_r$. These parameters as well as the sample data $x_i$ are used to run simulation $(i,r)$.

Let $S(x_i,\theta_r)$ (or shorthand, $S_{ir}$) be the simulation output for sample $i$ with uncertainty parameters equal to $\theta_r$. Note that these outputs $S(x_i,\theta_r)$ can be scalars, a classification category, vectors, or any other object. There is no need for these outputs to be the same as what we are trying to predict, $y_i$. We simply assume that given two such outputs, say $S_{ir}$ and $S_{jr}$ for samples $i$ and $j$, we have a way to measure the similarity between them. Let this similarity be given by $z(i,j,r)$. We leave it up to the user to define this function in general (a concrete procedure, for simulations using ordinary differential equations, is given in the Supplementary information).

Finally, the similarity $K(i,j)$ between two samples $i$ and $j$ is the average similarity across the simulation runs:

$$K(i,j) = \frac{1}{R}\sum_{r=1}^{R} z(i,j,r)$$

The above SimKern kernel matrix generation procedure is implemented in Python and is fully described in the Supplementary information.

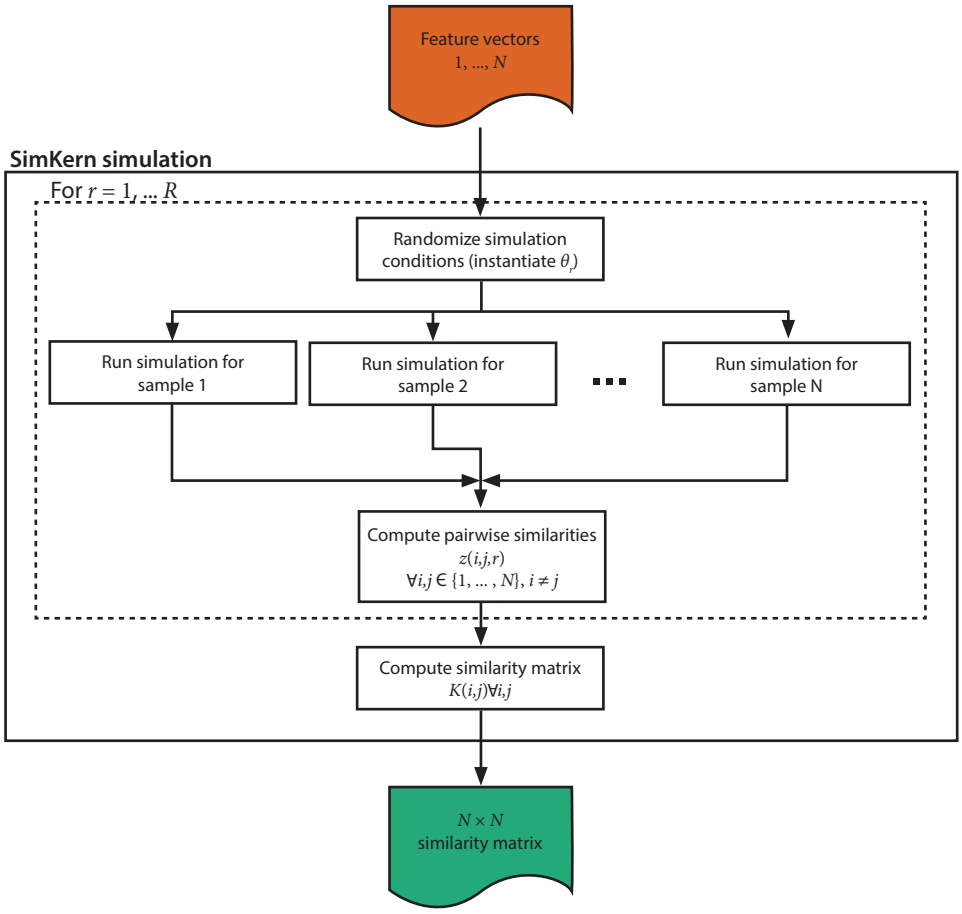



**Figure 2.** Creation of the similarity matrix for use downstream in the machine learning.

### *Machine learning comparisons procedure*

Figure 1 shows a schematic of the differences in the data processing and machine learning steps for Standard ML and SimKern ML. We compare standard feature-based ML algorithms (orange/top: linear support vector machine (SVM), radial basis function (RBF) SVM, and random forest (RF)) with simulation kernel based methods (green/bottom: kernelized SVM and kernelized RF). We also include results for 1-nearest neighbor (NN) and kernelized 1-nearest neighbor (SimKern NN). As NN-type algorithms are arguably the simplest non-trivial ML algorithms, including these algorithms allows us to understand the distinct contributions of ML algorithm sophistication and simulation-based kernels.

Since we can generate as many samples as we wish, we train the models and tune the hyperparameters on training and validation datasets which are distinct from the final testing set on which we compute prediction performance metrics (see section *Performance metrics*). The ground truth simulation generates one dataset which is then split into three parts (train/validation/test) using the standard proportions 50%/25%/25%[12, p.222].

SVM[13] and NN algorithms are dependent on feature scaling, therefore, features are standardized to the interval  by subtracting the minimum value and scaling by the range. Categorical features are dummy-coded for SVM and NN algorithms.

Each ML algorithm is trained on the training data for many hyperparameter configurations and the configuration with the best fit on the validation data is selected. The model given the selected configuration is applied on the test set to compute the performance metrics. See Alg. 1 in the Supplementary information for the details of training, hyperparameter tuning, and testing procedures.

To investigate the performance of simulation-based kernels in scenarios with less data for training, we consider five scenarios in which we train the algorithms on subsamples comprising of the training data. The subsampling percentages are chosen differently per model to highlight the interesting regions of curves that display the performance versus training set size. Table S3 reports the subsampling percentages per model.



*Performance metrics*

For each of the simulation models, we estimate the generalization performance of an ML algorithm in test data, i.e. data unused for model training, as performance estimates on training data are of little practical value[12, p.230].

The learning tasks per model are either classification or regression. For classification, we consider prediction accuracy, which is defined as

$$\text{Accuracy} = \frac{true\ classification\ count}{total\ number\ of\ samples} = \frac{TP+TN}{TP+TN+FP+FN}$$

where *TP, TN, FP, FN* are the counts of true positives, true negatives, false positives, and false negatives, respectively.

For regression, we consider the coefficient of determination $R^2$, which is defined as

$$R^2 = 1 - \frac{sum\ of\ squared\ prediction\ error}{sum\ of\ squares} = 1 - \frac{\sum_i(\hat{y}_i - y_i)^2}{\sum_i(\bar{y} - y_i)^2}$$

where $y_1$ is the outcome for sample *i*, $\hat{y}_i$ is the predicted outcome for sample *i*, and $\bar{y}$ is the sample mean of the outcome.

To attain a reliable estimate of the generalization performance, we consider the average test data performance in ten repetitions of a train/validation/test analysis, i.e. repeating training and hyperparameter tuning each time.

*Standard ML vs. SimKern ML comparison*

For each model, we produce a box plot and/or a line plot that show algorithm performance versus training dataset size for the various ML algorithms in both algorithm groups, Standard ML and SimKern ML.

1. Box plots display results for each algorithm separately for the Standard ML (linear SVM, RBF SVM, RF, NN) and SimKern ML algorithms (SimKern SVM, SimKern RF, SimKern NN). The horizontal lines indicate the sample median, the boxes are placed between the first and third quartile ($q_1$,$q_3$). Outliers are defined as samples outside $[q_1-1.5(q_3-q_1),q_3+1.5(q_3-q_1)]$ and are indicated by crosses.
2. Line plots further condense the findings by displaying the median performance metric of the best performing Standard ML and SimKern ML algorithms, excluding NN algorithms in both cases. The best performing algorithm is defined as the algorithm that most frequently produces the highest median performance metric over all five training dataset subsamples. Lines are interpolated for visual guidance.

*Sensitivity analysis*

To investigate possible factors affecting the SimKern algorithms' prediction performance, we run the following sensitivity analyses:

**Varying prior knowledge**

1. Radiation model: we examine the results for two kernels which represent different levels of prior knowledge. Both cases utilize the same SimKern simulation, but the higher quality kernel uses the dynamics of only the compartments of the ODE set that are used in the classification of the samples in the initial ground truth simulation. The lower quality kernel uses all ODE equations, therefore not emphasizing the most important ones[14].

**Varying simulation parameter noise/bias**

2. Network flow model: we generate two kernels for the network flow model. These kernels differ in the number of arc costs that are perturbed and the size of the perturbations (full details in Supplementary information).
3. Flowering time model: along with the model that generates the baseline kernel, we study one less noisy, one noisier, and one biased version of the SimKern simulation. The baseline SimKern simulation uses multiplicative Gaussian noise on 34 of the rate parameters, using a mean of 1 and a standard deviation of 0.2. The less noisy model uses stdev=0.1 and the noisier model uses stdev=0.4. For a more radical, and non-centered, departure from the true rate parameters, we also run a model where we multiply each of the same 34 rate parameters with a random variable chosen uniformly from the discrete set {0.01,1,5,10}.

**Varying the number of simulation trials, *R***

4. Network flow model: we analyze the effect of additional simulation trials on the prediction performance. We compare the prediction performance of SimKern algorithms when using a similarity kernel based on $R = 3$ simulation trials to the final kernel based on $R = 10$ trials. Furthermore, we track the convergence of the kernel matrix over $R = 10$ trials.

## Results

The general theme that emerges is that, for small training dataset sizes, the methods using the SimKern kernel outperform the Standard ML methods. For larger training sizes, however, the standard methods either approach the SimKern methods or exceed them, depending on the quality of the kernel.

For the radiation model, we see exactly this general pattern (Figure 3). For small training sizes (up to 50 samples), the SVM with the SimKern kernel dominates. We can attribute much of the performance gain to the similarity kernel itself given that the NN algorithm using the same similarity kernel also dominates over the no-prior-knowledge methods for all training sizes shown. The increase in accuracy by the Standard ML algorithms does not yet show signs of saturation by 500 training samples. These box plots are summarized by line plots in Figure 7 (left), which also displays the results of the lower quality SimKern kernel, which was made with the same simulations but without focusing on the most relevant ODEs for the kernel matrix computation.



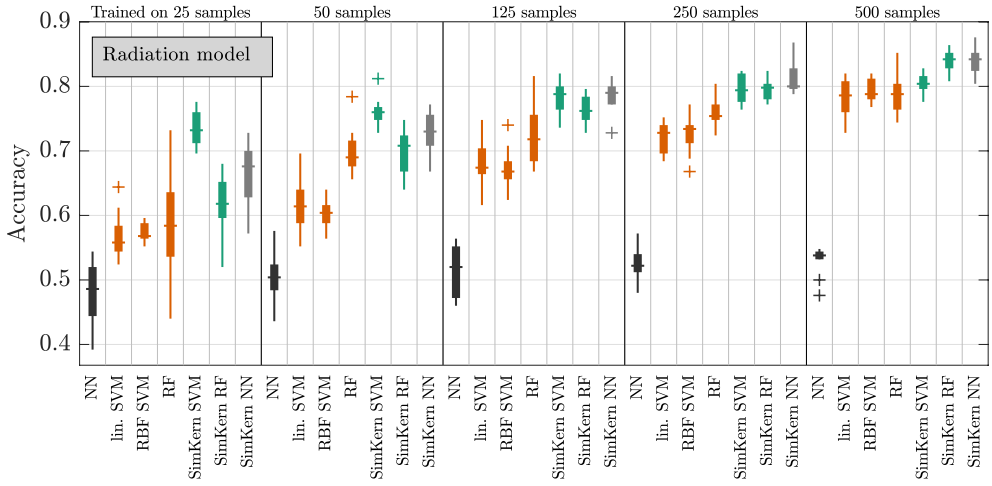



**Figure 3.** Machine learning results for the radiation cancer model. NN = nearest neighbor, RF = random forest, SVM = support vector machine, RBF = radial basis function.

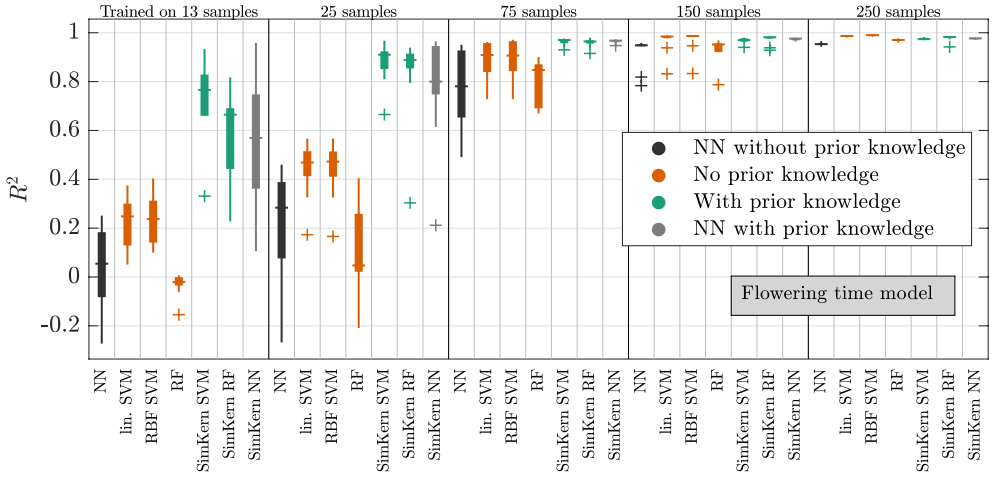



**Figure 4.** Machine learning results for the flowering time model. NN = nearest neighbor, RF = random forest, SVM = support vector machine, RBF = radial basis function.

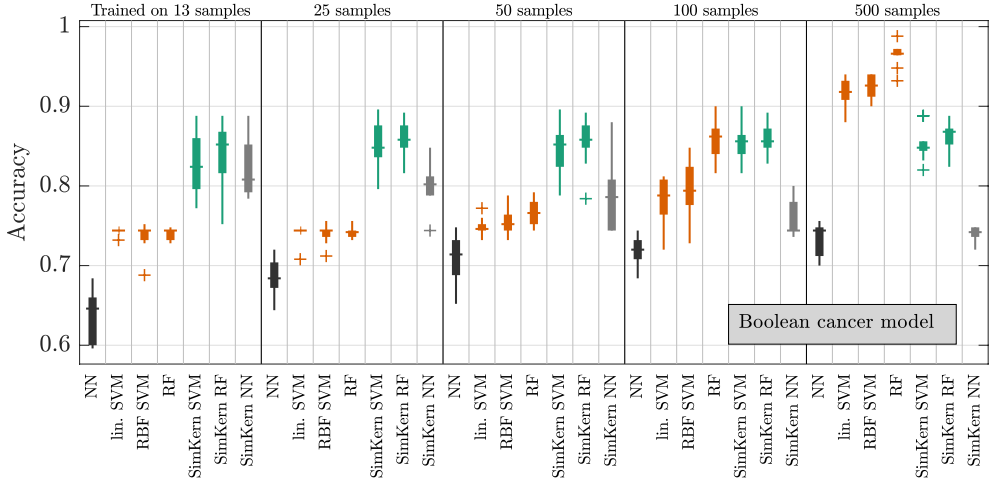



**Figure 5.** Machine learning results for the Boolean cancer model. NN = nearest neighbor, RF = random forest, SVM = support vector machine, RBF = radial basis function.

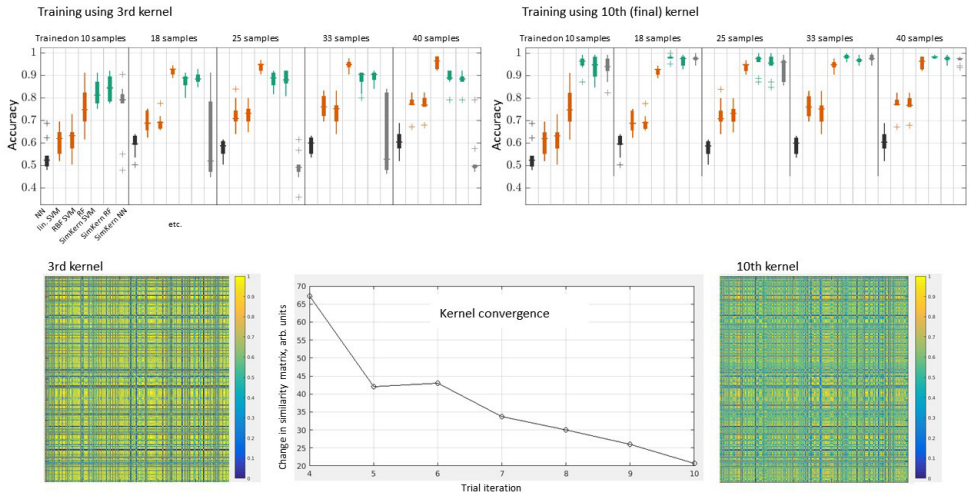



**Figure 6.** Varying simulation trials experiments for the network flow model. The upper two box plots compare the learning accuracy for a kernel from the third of $R = 10$ trials versus the final kernel. The kernels themselves are displayed with the same color scale below, and centered at bottom displays the convergence of the kernel (measured using the Frobenius matrix norm) over the ten trials. NN = nearest neighbor, RF = random forest, SVM = support vector machine, RBF = radial basis function.

The results of the flowering time model, which also display the clear dominance of SimKern learning for small training data set sizes, show a trend of decreasing variance in predictive performance with increasing training sizes (Figure 4). SimKern learning is strongly dominant up to 75 training samples, after which the two learning styles converge to $R^2 \approx 1$. Another view of the improvement offered by the SimKern method for small training size set sizes is shown by plotting the predicted flowering times versus the actual flowering times, Figure S9.

The sensitivity results obtained by increasing the variance of the (centered) Gaussian noise that was applied to the flowering model's rate parameters display a robustness to these deviations (Figure 8, upper green curves and Gaussian box plots). However, the non-centered noise perturbation analysis shows a clear drop in ML accuracy (Figure 8, dark green dotted line and dark green box plot). With enough training data, all SimKern kernels, including the ones with heavy noise, achieve an $R^2$ above 0.95. We call such kernels *sufficient*.

In contrast, the Boolean cancer model kernel is based on a model reduction with additional uncertainty and produces what we call a *biased* kernel. There, the SimKern approach produces an accuracy that initially dominates but quickly plateaus to around 85% and is overtaken by no-prior-knowledge methods when more training data is available (Figure 5). The fact that the kernel learning barely improves with additional data implies that the feature space induced by the simulation kernel is simple enough to be learned by a small amount of samples[15]. The kernelized NN method gets worse with more samples, and in general is worse than the other SimKern algorithms, which indicates that the space induced by the biased kernel is less cleanly separable compared to the flowering model case. Above 100 training samples, the no-prior-knowledge RF method is the superior technique.

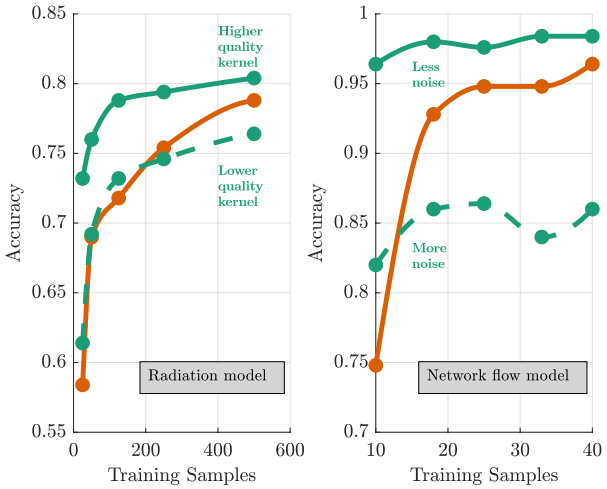



**Figure 7.** Varying prior knowledge experiments for the radiation model (left) and varying parameter noise experiments for the network flow model (right). Performance metrics of SimKern ML based on simulations with less and more prior knowledge (green) and Standard ML (orange). For each line, the best performing algorithm of SimKern ML or Standard ML is selected (see section *Standard ML vs. SimKern ML comparison*). Note, the waviness of the less noise case for the network flow model is an artifact of how the data from the box plots was converted into a line plot; the full data, Figure S7, reveals a flat relationship.

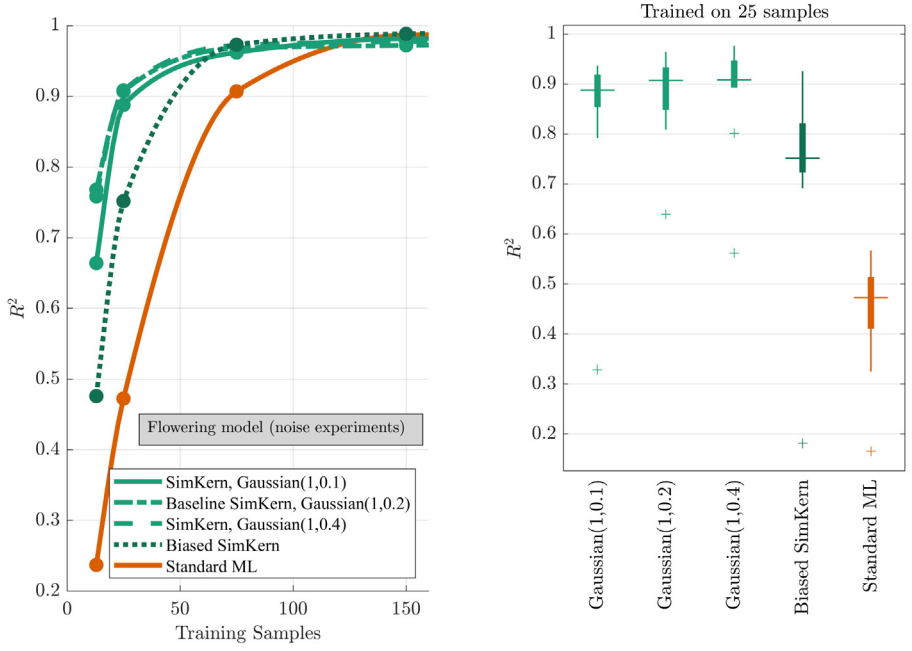



**Figure 8.** Varying simulation parameter noise/bias experiments for the flowering time model. SimKern ML based on simulations with varying parameter noise (green), with parameter bias (dark green), and Standard ML (orange). Left: performance metrics of SimKern ML (green) and Standard ML (orange) trained on up to 150 samples. For each line, the best performing algorithm of SimKern ML or Standard ML is selected (see section Standard ML vs. SimKern ML comparison). Right: performance metrics box plots for the 25 training sample case.

For the network flow problem we evaluate two separate kernels (Figure 7, right) based on different levels of noise in the SimKern simulation: the kernel based on a less noisy SimKern simulation dominates throughout, but even the kernel based on a noisier SimKern simulation is still useful in the very small training set size range. It is doubtful whether one can make general statements about how good a simulation needs to be in order to yield a useful kernel. However, the intuition that the simulations need only discover the similarity of samples, while not necessarily providing accurate (hence directly useful) simulation results, is described in Figure S8.

When comparing the individual Standard ML algorithms to the SimKern ML algorithms based on the noisier SimKern simulation (Figure S7), Standard RF eventually dominates. When comparing algorithms within the Standard ML group, RF is the dominant Standard ML algorithm for the network flow model (Figure S7) as well as for the Boolean cancer model (Figure 5). For these models, the dominance of RF is likely related to the discrete characteristics of the underlying models.



The quality of a simulation-generated kernel also depends on the number of trials that are used to compute the kernel. Figure 6 displays both the convergence of the kernel (bottom) and the improved learning accuracy from the further converged kernel (top), for the less noisy network flow case. We see that the earliest kernel written, kernel three (we chose to not determine similarity kernels below $R = 3$), performs noticeably worse than the final kernel. We can also visually observe the differences in the kernels by plotting the 500×500 kernels (Figure 6, bottom left and right). The kernel convergence plot is obtained by taking the Frobenius norms of the difference of the kernel matrices of iteration $i - 1$ and i, until $i = R(=10)$.

## Discussion

We introduce simulation as a pre-processing step in a machine learning pipeline, in particular as a way to include expert prior knowledge. One can consider simulation as a technique which regularizes data or as a specialized feature extraction method. In either view, the SimKern methodology offers a decomposition of an overall ML task into two steps: similarity computation followed by predictive modeling using the pairwise similarities. This decomposition highlights that to improve the performance of an ML model one can direct efforts into determining better similarity scores between all samples. This is in contrast to the more commonly heard call for "more data" to achieve better ML results. Of course, more samples are always desirable, but here we show that, particularly in limited data settings, sizable performance gains can come from high quality similarity scores.

The decomposition of simulation and machine learning steps also points out their individual contributions. The simulation-based kernel structures the space in which the samples live (or more technically, the dual of the space[16]), and ML finds the patterns in this simplified space. We see that in order to improve machine learning performance we can either improve the kernel or increase the number of samples to better populate the space. For the cases shown here, custom similarity measures show large improvements especially in limited data settings (up to a 20% increase in classification accuracy and a 2.5 fold increase in $R^2$, depending on the case and the amount of training data used). One could also use the output of the simulations as features for machine learning rather than the additional kernelization step that we employed. Using the simulation outputs directly is related to the field of model output statistics from weather forecasting, where low level data from primary simulations are used as inputs to a multiple regression model which outputs human-friendly weather predictions[17]. In our case, we opted for kernelizing the simulation outputs to highlight the fundamental concept of similarity and because a similarity computation is natural when the output of the simulations is a set of time varying entities, e.g., in the case of ODEs.

Similar in spirit to SimKern, although differing in details, combining simulation and machine learning has been used in physics to predict object behaviour[18,19]. Simulation results are used to train networks to "learn" the physics. Varying the simulation conditions during training, called *domain randomization*, is used to improve model generalization[20]. Inversely to the SimKern approach to exploit simulation to enhance ML algorithms, machine learning is also used to correct the inputs to physics simulations[21], an idea which is also pursued in the context of traffic prediction[22].

A novel potential application of the SimKern methodology, one that the authors are currently investigating, involves the prediction of peptides (chains of approximately nine amino acids) binding to a given human leukocyte antigen (HLA) class 1 allele. Current technologies (e.g.,[23]) predict if a given peptide will bind to a given HLA allele using properties of the amino acids but without using 3D details of the chemical structure of the peptide or information on the structural binding of the peptide and HLA molecule. Computational predictions of binding are considered too difficult at the present time due to the sensitivity of the structural conformations to the detailed chemistry of peptides and the non-covalent interactions[24]. Nevertheless, simulations could be used to generate similarity scores between peptides, and then the supervised binding data can be used to train a kernelized classification algorithm.

Finally, the use of a SimKern kernel need not be an all-or-nothing decision, since two or more kernels can be combined to yield a single kernel. This allows one to explore the combination of "standard" kernelized learning (using uninformed kernels such as linear or RBF) with a SimKern kernel. In the case of a weighted linear sum as the method of kernel combining, one can optimize the weighting vector as part of the training procedure[7]. Combining kernels allows one to mix traditional feature-based machine learning (which we called Standard ML above) with prior knowledge similarity matrix-based learning.

## Conclusions

It remains to be seen which approaches will be the most fruitful as we make our way towards personalized cancer medicine. Direct testing of chemotherapeutic agents on biopsied patient tissues is a straightforward and promising "hardware-based" approach[25]. In the machine learning realm, expert feature selection may turn out to be more feasible than the simulation-based kernel methods described in this report. A key question is: can we make simulation-based kernels that—although almost certainly biased—will still be useful (see, e.g., Figure 7)? Progress in detailed biological simulation, such as the full simulation of the cell cycle of the bacterium *Mycoplasma genitalium*[26], the OpenWorm project[27], and integrated cancer signaling pathways for predicting proliferation and cell death[28] offer some encouragement, but cancer influences human biology at all levels, from minute phosphorylations to immune system rewiring. It is thus by no means clear if we are close to simulations that can be useful in this context. However, the magnitude of the problem—both in economic terms and for the number of future patients at stake—suggests pressing forward on all fronts that display conceptual promise.

## References


1. Allan Balmain, Joe Gray, and Bruce Ponder. The genetics and genomics of cancer. *Nature genetics*, **33**:238, 2003.
2. E Melo Felipe De Sousa, Louis Vermeulen, Evelyn Fessler, and Jan Paul Medema. Cancer heterogeneity-a multifaceted view. *EMBO reports*, **14**(8):686–695, 2013.
3. R Fisher, L Pusztai, and C Swanton. Cancer heterogeneity: implications for targeted therapeutics. *British journal of cancer*, **108**(3):479–485, 2013.
4. Haitham Mirghani and Pierre Blanchard. Treatment de-escalation for HPV-driven oropharyngeal cancer: Where do we stand? *Clinical and Translational Radiation Oncology*, 2017.
5. Alexander S Hauser, Sreenivas Chavali, Ikuo Masuho, Leonie J Jahn, Kirill A Martemyanov, David E Gloriam, and M Madan Babu. Pharmacogenomics of gpcr drug targets. *Cell*, **172**(1-2):41–54, 2018.
6. Diego Chowell, Luc GT Morris, Claud M Grigg, Jeffrey K Weber, Robert M Samstein, Vladimir Makarov, Fengshen Kuo, Sviatoslav M Kendall, David Requena, Nadeem Riaz, et al. Patient hla class i genotype influences cancer response to checkpoint blockade immunotherapy. *Science*, **359**(6375):582–587, 2018.
7. Bernhard Schölkopf, Koji Tsuda, and Jean-Philippe Vert. *Kernel methods in computational biology*. MIT press, 2004.
8. Ján Eliaš, Luna Dimitrio, Jean Clairambault, and Roberto Natalini. The p53 protein and its molecular network: modelling a missing link between dna damage and cell fate. *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics*, **1844**(1):232–247, 2014.
9. Felipe Leal Valentim, Simon van Mourik, David Posé, Min C Kim, Markus Schmid, Roeland CHJ van Ham, Marco Busscher, Gabino F Sanchez-Perez, Jaap Molenaar, Gerco C Angenent, et al. A quantitative and dynamic model of the arabidopsis flowering time gene regulatory network. *PloS one*, **10**(2):e0116973, 2015.
10. David PA Cohen, Loredana Martignetti, Sylvie Robine, Emmanuel Barillot, Andrei Zinovyev, and Laurence Calzone. Mathematical modelling of molecular pathways enabling tumour cell invasion and migration. *PLoS computational biology*, **11**(11):e1004571, 2015.
11. D. Bertsimas and J. Tsitsiklis. *Introduction to linear optimization*. Athena Scientific, 1997.
12. Trevor Hastie, Robert Tibshirani, and JH Friedman. *The elements of statistical learning*, volume 2. Springer-Verlag New York, 2009.
13. Asa Ben-Hur and Jason Weston. A user's guide to support vector machines. *Data mining techniques for the life sciences*, pages 223–239, 2010.
14. Dana Ferranti, David Krane, and David Craft. The value of prior knowledge in machine learning of complex network systems. *Bioinformatics*, **33**(22):3610–3618, 2017.
15. Léon Bottou, Corinna Cortes, and Vladimir Vapnik. On the effective vc dimension. Technical Report bottou-effvc.ps.Z, Neuroprose, 1994. Also available on http://leon.bottou.org/papers.
16. Sun Yuan Kung. Kernel methods and machine learning. Cambridge University Press, 2014.
17. Harry R Glahn and Dale A Lowry. The use of model output statistics (MOS) in objective weather forecasting. *Journal of applied meteorology*, **11**(8):1203–1211, 1972.
18. Adam Lerer, Sam Gross, and Rob Fergus. Learning physical intuition of block towers by example. *arXiv preprint arXiv*:1603.01312, 2016.
19. Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In *Advances in neural information processing systems*, pages 127–135, 2015.

20. Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 23–30. IEEE, 2017.
21. Karthikeyan Duraisamy, Ze J Zhang, and Anand Pratap Singh. New approaches in turbulence and transition modeling using data-driven techniques. In *53rd AIAA Aerospace Sciences Meeting*, page 1284, 2015.
22. Muhammad Shalihin Bin Othman and Gary Tan. Predictive simulation of public transportation using deep learning. In *Asian Simulation Conference*, pages 96–106. Springer, 2018.
23. Morten Nielsen, Claus Lundegaard, Thomas Blicher, Kasper Lamberth, Mikkel Harndahl, Sune Justesen, Gustav Røder, Bjoern Peters, Alessandro Sette, Ole Lund, et al. Netmhcpan, a method for quantitative predictions of peptide binding to any hla-a and-b locus protein of known sequence. *PloS one*, **2**(8):e796, 2007.
24. Prattusha Kar, Lanie Ruiz-Perez, Mahreen Arooj, and Ricardo L Mancera. Current methods for the prediction of t-cell epitopes. *Peptide Science*, **110**(2):e24046, 2018.
25. Joan Montero, Kristopher A Sarosiek, Joseph D DeAngelo, Ophélia Maertens, Jeremy Ryan, Dalia Ercan, Huiying Piao, Neil S Horowitz, Ross S Berkowitz, Ursula Matulonis, et al. Drug-induced death signaling strategy rapidly predicts cancer response to chemotherapy. *Cell*, **160**(5):977–989, 2015.
26. Jonathan R Karr, Jayodita C Sanghvi, Derek N Macklin, Miriam V Gutschow, Jared M Jacobs, Benjamin Bolival Jr, Nacyra Assad-Garcia, John I Glass, and MarkusWCovert. A whole-cell computational model predicts phenotype from genotype. *Cell*, **150**(2):389–401, 2012.
27. Balázs Szigeti, Padraig Gleeson, Michael Vella, Sergey Khayrulin, Andrey Palyanov, Jim Hokanson, Michael Currie, Matteo Cantarelli, Giovanni Idili, and Stephen Larson. Openworm: an open-science approach to modeling caenorhabditis elegans. *Frontiers in computational neuroscience*, **8**:137, 2014.
28. Mehdi Bouhaddou, Anne Marie Barrette, Rick J Koch, Matthew S DiStefano, Eric A Riesel, Alan D Stern, Luis C Santos, Annie Tan, Alex Mertz, and Marc R Birtwistle. An integrated mechanistic model of pan-cancer driver pathways predicts stochastic proliferation and death. *BioRxiv*, page 128801, 2017.

## Supplementary information

*Ground truth and SimKern simulations*

The simulation framework, which handles the generation of ground truth data as well as the SimKern module which performs the simulations and computes the similarity matrix, is written in Python, and supports simulation models written in MATLAB, Octave, and R. It uses text file communication so it could be easily adapted to simulations written in other languages. The Python package, SimKern, is available at github: **https://github.com/davidcraft/SimKern**. *We refer to the ground truth simulation as SIM0 and the SimKern simulations as SIM1. This naming convention is also reflected in the Python code base.*
The various code modules are summarized in Table S1.

**Table S1.** Code module descriptions.

| Name | Functionality | Requires | Language |
|---|---|---|---|
| Groundtruth dataset generation ("SIM0") | Generates datasets (features and known outcomes) with user-selected number of samples | The simulation ("SIM0") model (*.t file) | Python (simulation models though are in Matlab, octave, or R) |
| SimKern ("SIM1") | Handle running families of simulations, aggregating results and forming the similarity kernel | Feature vectors that are used to simulate each feature ("genome key" files), and the master *.u file that contains the stochasticity information $\boldsymbol{\theta}$ | Python (as above) |
| Machine Learning Comparison | Tune and train models with all machine learning algorithms on various dataset sizes for comparison | Sample features for standard machine learning, sample similarity matrix for kernelized learning, and sample outcomes | Matlab (also available in the SimKern python repository, but Matlab version used for the results in the paper) |

*Ground truth data generation procedure: SIM0*

A simulation model file used to create a ground truth dataset has the suffix .t. A file used to create the SimKern family of simulations is suffixed with .u (see next section). These model files are in the language of the system used to run the simulations and have entities that are set off by dollar signs. These entities are the parameters to vary from one sample to the next, for the ground truth dataset generation, or from one trial to the next, for the SimKern generation.

As an example, if different samples may have different values for a rate parameter called $k_1$, a line in the simulation file could read:
k1 = $gauss(8,2, name='decayConstant1')$;
The Python code will replace the text set off by the dollar signs with a random variable drawn from a Gaussian distribution with mean 8 and standard deviation 2. In the file storing the sample features that gets written, this feature will be named decayConstant1. This same style is used for both SIM0 and SIM1. The distributions that are allowed, and more usage details, are given in the manual on the SimKern github repository.

If the simulation package to use is MATLAB, the Python package allows a direct process hook via a MATLAB-Python API provided by MathWorks. This speeds up the overall runtime by not requiring the expensive startup time of MATLAB for every run.

Let $N$ be the number of samples we generate for the SIM0 dataset. Let the feature vectors (the parameters that make the samples different from each other) be given by the vectors $x_i, i = 1, ..., N$. Each $x_i$ vector is a vector of length $p$, where we are following the standard machine learning notation where $p$ equals the number of features. Let $y_i$ denote the outcome of the simulation, which could be a category (e.g. alive or dead) or a real number. Since we generate these outputs via a simulation, viewing that simulation as a function $S^0$ we can write $y_i = S^0(x_i)$. The ground truth data generation procedure is depicted in Figure S1.

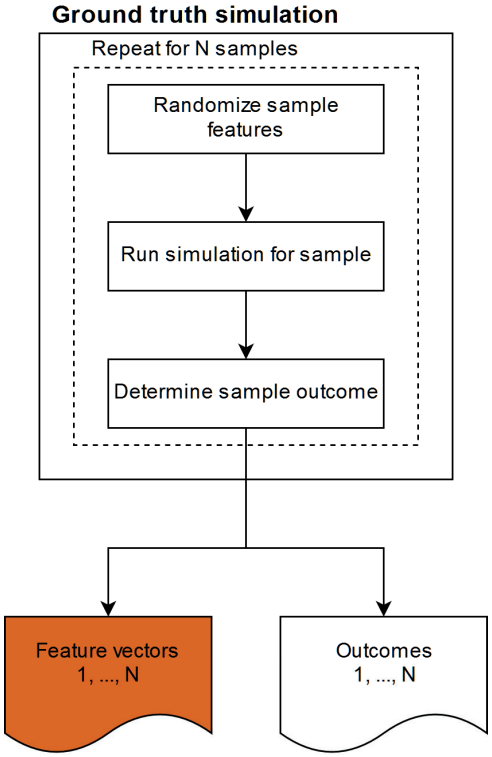



**Figure S1.** Ground truth data generation procedure, SIM0.

The *x* data get written to Sim0Genomes.csv and the data to Sim0Output.csv (the term genome is used since the use case that provides the motivation for this software is machine learning for biological systems where the feature vector is based on genomics). Separate files, called genome keys, are written out for each sample for use in the SIM1 runs.

*Similarity kernel generation: SIM1*

The main document describes the similarity matrix computation.

The python software handles writing out and running the individual (*i, r*) run files, using the .u file as the template. This .u file must reference another file which specifies the parameters from the SIM0 run that make each individual sample *i* distinct. This file is called genome1_key (the "1" is replaced automatically by the SIM1 python code with the sample number *i*). The output of this procedure is the similarity matrix, given in a file called SimilarityMatrixfinal.csv. A similarity matrix is also written after every trial (from the third trial onward; similarity matrices before the third trial are considered not converged yet and so are not written out).

*Similarity as measured by closeness of ODE solutions*

A typical setting for a SIM1 run will be the simulation of a set of ordinary differential equations (ODEs). In this context, the similarity between population members *i* and *j*, for simulation *r*, can be a measure of how close the overall time dynamics for *i* are to the time dynamics of *j*, e.g., represented by the mean squared error over discrete time points. More specifically, assume the ODE simulation contains *E* different entities (e.g. protein levels), in other words *E* ODEs. Let us further assume that the simulation program outputs the levels of these entities at a given set of times, $t_1,t_2,...,t_K,...,t_T$. Let $L_r^i$ be the level of ODE entity *e* at time $t_K$, for population member *i* under simulation *r*. Since the ODE equations may be of different magnitudes, we will normalize each pair being compared by the maximum level that either ever takes over the time course (we are implicitly assuming the ODEs solutions are always non-negative, this would have to be modified for negative levels). For the pair of samples (*i, j*) and for entity *e* in simulation run *r*, the maximum value *M* is given by:

$$M(i,j,e,r) \;=\; max[max_k L_r^i(e,k), max_k L_r^j(e,k)]$$

With these definitions, we can write

$$z(i,j,r) \;=\; 1 \;-\; \frac{1}{E*T}\sum_{e=1}^{E}\sum_{k=1}^{T}\left(\frac{L_r^i(e,k) - L_r^j(e,k)}{M(i,j,e,r)}\right)^2$$

Finally, in addition to normalizing the ODE solutions to a maximum value of 1, the user may want to weight the different entities *e* to express the prior knowledge that some entities are more important for similarity considerations than others. Let $0 \leq w_e \leq 1$ be user-defined weights and then we have:

$$z(i,j,r) = 1 - \frac{1}{E * T} \sum_{e=1}^{E} w_e \sum_{k=1}^{T} \left( \frac{L_r^i(e,k) - L_r^j(e,k)}{M(i,j,e,r)} \right)^2$$

## Machine learning details

*Machine learning algorithm comparisons procedure*

Machine learning (ML) was conducted in MATLAB (MathWorks, Natick, MA, USA) using the libSVM package for all SVM models[1]. The python SimKern codebase also provides routines for the machine learning runs. Alg 1 outlines the experimental design to tune the hyperparameters and then estimate performance metrics for each ML algorithm. Although the algorithm initially splits a dataset into three pieces—50% for training, 25% for validation, and 25% for final accuracy assessment—the training subset is further subsampled to assess how accuracy depends on the amount of training data for the various models and machine learning algorithms. The same experiment is repeated for each dataset. The procedure is outlined below and explained in detail in the subsequent subsections.

```
load data of the ground truth data simulation;
load similarity matrix of the SimKern simulation;
shift and rescale features to [0, 1];
dummy-code categorical features for SVM algorithms;
for repetition i = 1 : 10 do
    randomly sample 50% of all rows as training data (stratify samples if it is a
     classification problem);
    randomly sample 25% of all remaining rows as validation data (stratify samples
     if it is a classification problem);
    assign the remaining rows as test data;
    foreach subsampling percentage s ∈ {s_1, s_2, ..., s_S} do
        randomly subsample s of all training rows as training data (stratify samples
         if it is a classification problem);
        foreach algorithm a ∈ A do
            foreach hyperparameter configuration h_a ∈ H_a do
                train algorithm a with hyperparameter configuration h_a on training
                 data;
                predict outcomes for validation data;
                compute performance metric on validation data predictions;
            end
            select hyperparameter configuration h*_a with best validation performance
             metric;
            select algorithm a trained with hyperparameter configuration h*_a;
            predict outcomes for test data;
            compute performance metric on test data predictions;
        end
    end
end
```

**Alg. 1.** Experimental design to estimate ML performance (this algorithm is executed independently on each dataset). *A* is the set of ML algorithms used. *s1* subsampling percentages vary by model in order to home in on the most relevant part of the curve which represents accuracy versus amount of training data, see Table S3.

*Stratification*

For the classification models, the data is split while approximately stratifying for classes. Stratification of classes in training, validation, and test data ensures stability in the estimation process. Consider the case where random sampling led to an unusual distribution of classes in training and validation data. Consequently, the test data would very likely have a class distribution different than the training data. Classifiers not correcting for class imbalance (default RF and default SVMs) that are trained on this training data would perform worse on the test data. Since we want to estimate generalization performance, i.e. performance on the general population with a class distribution estimated by the class distribution in the full dataset, we stratify classes in training and test data.

*Hyperparameter tuning*

The performance of the studied ML algorithms is dependent on algorithm-specific hyperparameters (HP) whose optimal values for generalization performance are not known *a priori*. HPs are tuned by a grid search: for a selection of values per HP, the algorithm is trained on the training data and evaluated on the validation data for each possible HP combination. The HP combination with the best performance metric in the validation data is selected. Table S2 lists the HPs that are tuned, their ranges, and values on the search grid for each algorithm. Values are partly determined from existing literature or chosen experimentally. HPs not mentioned here are set to default values. Values for SVM parameters are partially taken from[2]. For RF, the number of trees is fixed at 100.

While Breiman (2001)[3] did not limit the number of terminal nodes in a tree, Duroux and Scornet (2016)[4] provide empirical evidence in favor of tuning. Therefore, we tune the maximal number of splits allowed in a tree. Tuning grid boundaries have been extended manually to reduce the number of cases where the tuning procedure selects HP values on the grid boundaries, which would suggests that better HP values might be found outside the grid.



**Table S2.** Hyperparameter tuning per algorithm. $C$ is the weight corresponding to training set error in the SVM objective. $\varepsilon$ (only used for SVM regression) determines the width of the margin enclosing the separating hyperplane in SVM regression. $\gamma$ is a parameter of the RBF kernel $K(x,y) = e^{(-\gamma\|x-y\|2)}$. *n. feat.* is the number of randomly sampled features compared at each split in a tree. n. splits is the maximal number of splits per tree, grid values exceeding the [1, ($n$ - 1)]interval are truncated to the boundary. $n$ is the number of training samples, $p$ is the number of features.

| Algorithm | HP | Range | Values on grid |
|---|---|---|---|
| linear SVM & SimKern SVM | $C$ | $[0, \infty]$ | $\{10^{-12}, 10^{-11}, ..., 10^{12}\}$ |
| | $\varepsilon$ | $[0, 1]$ | $\{10^{-5}, 10^{-4}, ..., 10^{-1}, 0.25, 0.5, 0.75, 1\}$ |
| RBF SVM | $C$ | $[0, \infty]$ | $\{10^{-12}, 10^{-11}, ..., 10^{12}\}$ |
| | $\gamma$ | $(0, \infty]$ | $\{10^{-15}, 10^{-14}, ..., 10^{1}\}$ |
| | $\varepsilon$ | $[0, 1]$ | $\{10^{-5}, 10^{-4}, ..., 10^{-1}, 0.25, 0.5, 0.75, 1\}$ |
| RF & SimKern RF | *n. feat.* | $[1, \infty]$ | $\{1, \lfloor(1+\sqrt{p})/2\rfloor, \lfloor\sqrt{p}\rfloor, \lfloor(\sqrt{p}+p)/2\rfloor, p\}$ |
| | *n. splits* | $[1, (n-1)]$ | $\lfloor\{0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.75, 1\}n\rfloor$ |

**Table S3.** Subsampling training percentages per model.

| Model | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ |
|---|---|---|---|---|---|
| Radiation | 5% | 10% | 25% | 50% | 100% |
| Flowering | 5% | 10% | 30% | 60% | 100% |
| Boolean | 2.5% | 5% | 10% | 20% | 100% |
| Network | 4% | 7% | 10% | 13% | 16% |

*Machine learning algorithms*

We compare standard machine learning algorithms that use the ground truth feature vectors (Standard ML algorithms) to ML algorithms that use the SimKern kernel matrix (*SimKern* ML algorithms), see Figure 2. For the Standard ML learning, we utilize three established ML algorithms: linear SVM[5], radial basis function (RBF) SVM, and random forest (RF)[3]. For SimKern learning, we use SVM with the similarity matrix as a custom kernel (note that for the SVM algorithm the kernel matrix, also known as the Gram matrix, has to be symmetric positive definite, which in all of our models is the case, and indeed is required by the libSVM software) and the random forest algorithm with the similarity matrix as the feature matrix input[6]. This random forest, called SimKern RF, classifies new samples according to their similarities with training samples.

Additionally, we compute nearest neighbor predictions to compare to the more advanced machine learning algorithms. For the Standard ML case, we use a 1-NN algorithm on the SIM0 feature vector. For the SimKern case, we use the label of the most similar distinct training sample according to the similarity matrix. We label this approach *SimKern NN*.

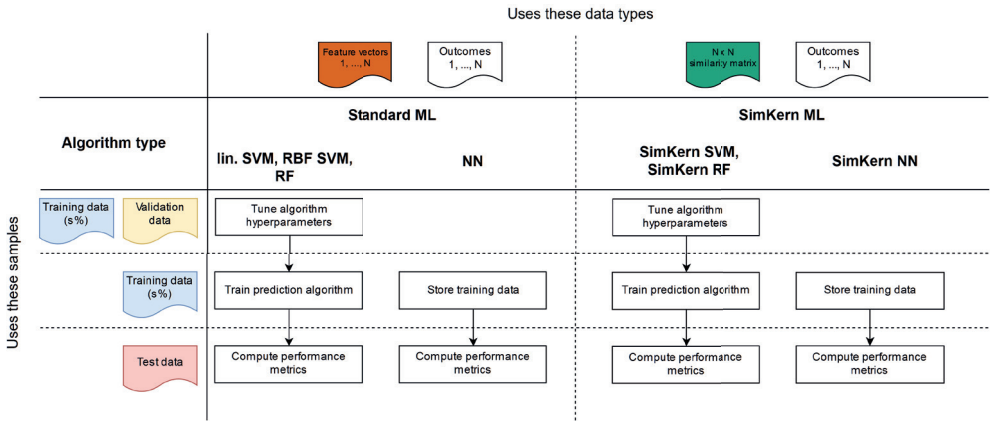



**Figure S2.** An overview of the data handling procedures for the various machine learning algorithms used. SVM=support vector machine, RBF=radial basis function, ML=machine learning, NN=nearest neighbors, RF=random forest.

## Model descriptions

Table S4 gives a summary of the machine learning problem sizes, number of features, and other attributes, for the four models.

**Table S4.** Numerical information for the four models. Class distribution per model for the ground truth (SIM0) dataset. Note that the Flowering model has continuous outcomes (i.e. flowering time) and the Boolean and Network models have only three classes. Classes (in order 1, 2, 3, 4) for the Radiation model are apoptosis, repaired and cycling, mitotic catastrophe, and quiescence. For the Boolean cancer model they are apoptosis, metastasis, and other. For the Network model they are simply which of the exit arcs the optimal solution flows through. $n$ is the number of samples generated for the SIM0 ground truth dataset, $p$ is the number of features in the ground truth dataset, and $R$ is the number of trials run in the SimKern step. *For the flowering model one of the features is a categorical variable of 19 classes, representing 19 different mutational states. Thus if one-hot encoded this would lead to an additional 19 features.

| Model | Class 1 | Class 2 | Class 3 | Class 4 | $n$ | $p$ | $R$ |
|---|---|---|---|---|---|---|---|
| Radiation | 27.5% | 23% | 44.2% | 5.3% | 1000 | 39 | 20 |
| Flowering | - | - | - | - | 500 | 35* | 5 |
| Boolean | 62.5% | 9.3% | 28.2% | - | 1000 | 37 | 20 |
| Network | 61.6% | 18.2% | 20.2% | - | 500 | 12 | 10 |



*Radiation model*

The radiation model is built up as four connected modules. We opt to not simulate the cell cycle and instead focus on the chain of events that happens after radiation damages a cell's DNA: DNA repair (modeled at a high level), p53-based transcription factor control, cell cycle arrest, and apoptosis, see Figure S3. Although highly simplified, this model recapitulates the idea that the inter-connected dynamics of these processes determine cell fate after radiation damage.

Tuning this model to reflect the behavior of an actual cell line is very large task, and probably not possible in any realistic way, since the genes (proteins) chosen to be in the model are but a small subset of the proteins involved in a DNA repair and cell cycle control cascade. However, even without validated rate constants chosen, the model provides a numerical instance of a complex system, based on known biology, where different modules (biochemical processes) are involved in determining the fate of a cell subject to an external stimulus. We hand tuned the parameters of the base model. There are many parameters to choose from, and our choices were from manual explorations which led to a set of parameters that led to diverse system behavior (some samples ending in apoptosis, others in cell cycle arrest, etc.).

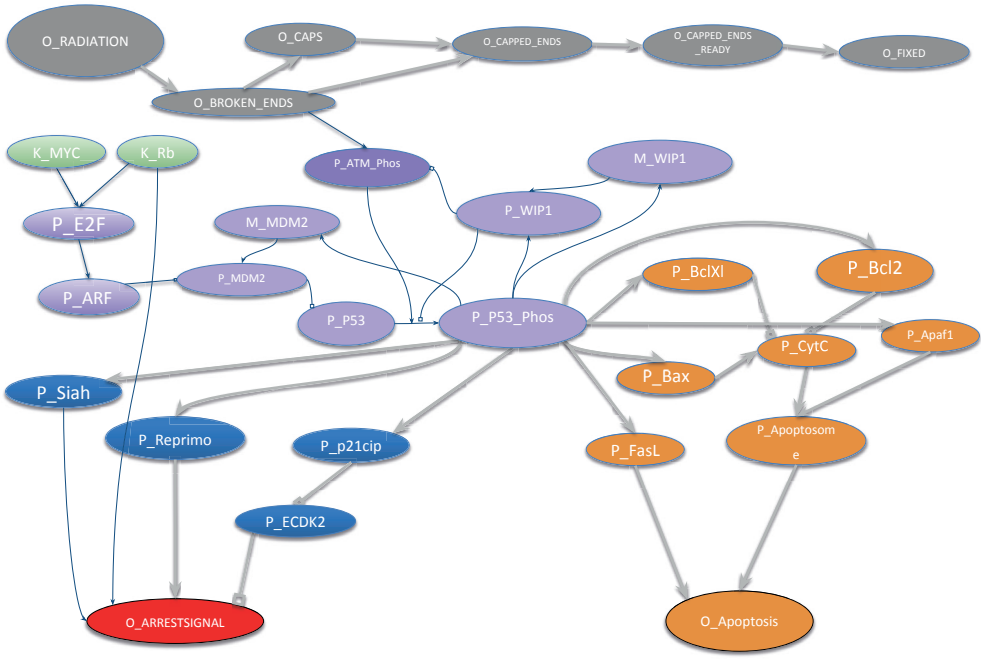



**Figure S3.** A model of entities and processes involved in cell fate decision following radiation. The gray nodes depict the process of DNA breakage and repair. DNA breaks send signals via ATM to the p53-MDM2-ARF module (purple), which in turn sends both apoptotic signals (orange) and cell cycle arrest signals (blue). The cancer genes MYC and Rb (green) are modeled as fixed parameters rather than time varying entities. The first letters of each oval have the following meanings: P = protein, M = mRNA, K = rate constant, O = other. Phos stands for phosphorylated.

The p53-MDM2 transcription regulatory control circuit comes from Eliaš et al. (2014)[7].
We use the single compartment version of the model, where the specific location of molecules (nucleus versus cytoplasm) is ignored. Radiation damage affects this circuit via the ATM kinase pathway, which increases the phosphorylation and hence stability of p53. p53 then goes on to be a transcription factor for apoptosis and cell cycle arrest genes.

Cell specific alterations (mutations, amplifications, deletions) for MYC, RB1, and p53 interact to influence how the p53-MDM2 circuit behaves, which in turn affects the behavior of the downstream processes of cell cycle arrest and apoptosis. The number of cell cycle controls in an eukaryotic cell is large. Rather than attempting to model most of them, we choose a few overlapping controls to create a model that creates a challenging machine learning problem.

Apoptosis is modeled as the competition between pro-apoptotic (BAX, FasL) and anti-apoptotic proteins (BCL-2, BCL-xl). Apoptosis occurs if the apoptosome is formed (a combination of cytochrome c and APAF-1, which together release caspases from the mitochondrial membrane) or via the extrinsic Fas/FasL pathway.

The detailed mathematical model is given next. In the ODE equations as written below, we use a generic "*k*" for ODE constants, to reduce clutter. For the full details, we refer the reader to the MATLAB code.

Phosphorylated nuclear p53 protein tetramerizes to form its active transcription factor state. For convenience we define the p53 tetramerized term as:

$$p53tt = (MUT_{p53} * pP53NucPhos^4)$$

The mutation coefficient $MUT_{p53}$ is a uniform random variable between 0 and 1, reflecting the idea that there are a large number of p53 mutations that potentially affect the tetramerization in varying ways.

The full ODE model is given here:

$$\dot{oRadiation} = -k * oRadiation$$

$$\dot{oBrokenEnds} = k * oRadiation - k * oBrokenEnds * oCaps$$

$$\dot{oCaps} = \min((k * oBrokenEnds), k_5) - k * oBrokenEnds * oCaps - k * oCaps$$

$$\dot{oCappedEnds} = k * oBrokenEnds * oCaps - k * oCappedEnds$$

$$\dot{oCappedEndsReady} = k * oCappedEnds - k * oCappedEndsReady$$

$$\dot{oFixed} = k * oCappedEndsReady$$

$$\dot{pP53Nuc} = k + k * pWIP1Nuc * \frac{pP53NucPhos}{+pP53NucPhos} - k * pMDM2Nuc * \frac{pP53Nuc}{k * pP53Nuc} - k * pATMNucPhos \frac{pP53Nuc}{k * pP53Nuc} - k * pP53Nuc$$

$$\dot{pMDM2Nuc} = k * mMDM2Nuc - pMDM2NUC - MUT_{arf} * k * pARF * pMDM2Nuc$$

$$\dot{mMDM2Nuc} = k + k * \frac{p53tt}{k^4 + p53tt} - k * mMDM2Nuc - k * mMDM2Nuc$$

$$\dot{pP53NucPhos} = k * pATMNucPhos * \frac{pP53Nuc}{k * pP53Nuc} - k * pWIP1Nuc * \frac{k * pP53NucPhos}{k + pP53NucPhos}$$

$$\dot{pWIP1Nuc} = k * mWIP1Nuc - kpWIP1Nuc$$

$$\dot{mWIP1Nuc} = k + k * \frac{p53tt}{k^4 + p53tt} - k * mWIP1Nuc - k * mWIP1Nuc$$

$$\dot{pATMNucPhos} = 2 * k * oBrokenEnds * \frac{k - pATMNucPhos}{k + \frac{k - pATMNucPhos}{2}} - 2 * k * pWIP1Nuc * \frac{ATMNucPhos^2}{k + pATMNucPhos^2}$$

$$\dot{pBcl2} = k * \frac{p53tt}{k + p53tt} - k * pBcl2$$

$$\dot{pBclXl} = k * \frac{p53tt}{k + p53tt} - k * pBclXl$$

$$\dot{pFasL} = k * \frac{p53tt}{k + p53tt} - k * pFasL$$

$$\dot{pBax} = MUT_{Bax} * \left( k * \frac{p53tt}{k + p53tt} - k * pBax \right)$$

$$\dot{pApaf1} = MUT_{Apaf1} * \left( k * \frac{p53tt}{k + p53tt} - k * pApaf1 \right)$$

$$\dot{pCytC} = k * \frac{1}{1 + e^{-k*pBax-k}} * k * (1 - \frac{1}{1 + e^{-k*pBcl2-k}}) * k * (1 - \frac{1}{1 + e^{-k*pBdXl2-k}}) - kpCytC - k * pApaf1 * pCytC^7$$

$$\dot{pApoptosome} = k * pApaf1 * pCytC^7 - k * pApoptosome$$

$$\dot{oApoptosis} = k * pFasL + k * pApoptosome - k * oApoptosis$$

$$\dot{pE2F} = MUT_{Rb} * MUT_{myc} - kpE2F$$

$$\dot{pARF} = MUT_{arf} \left( k1 * \frac{pE2F}{k + pE2F} - k2 * pARF - k * pARF * pMDM2Nuc \right)$$

$$\dot{pP21cip} = k * \frac{p53tt}{k + p53tt} - k * pP21cip$$

$$\dot{pECDK2} = k - \frac{k * pP21cip}{k + pP21cip} - (k * pECDK2)$$

$$\dot{pSiah} = MUT_{Siah} \left( \frac{k * p53tt}{k + p53tt} - k * pSiah \right)$$

$$\dot{pReprimo} = MUT_{Reprimo} \left( \frac{k * p53tt}{k + p53tt} - k * pReprimo \right)$$

$$\dot{oArrestsignal} = (see\ below)$$

The initial condition of the system is an externally applied radiation dose modeled by setting *oRadiation*(0) = 1 followed by an exponential decay. The only other non-zero initial condition is for ECDK2 since at time 0 we assume that there are no brakes on the cell cycle.

*Additional modeling notes*

Cells have many mechanisms to control cell growth and division. We choose to model just a few, and in a simplified manner, to get the flavor of the complexity. We split the control into two cases, one where the Rb gene is functioning (*Rb* = 1) and one where the Rb gene is impaired (*Rb* = 0). For the *Rb* = 0 case, a way to arrest cell growth is via the SIAH or Reprimo gene pathways. SIAH and Reprimo are activated by a functioning p53 danger signal pathway, and we model their effect on the arrest signal as additive. Thus:

*Case Rb = 0:*

$$oArrestsignal = \frac{1}{1 + e^{ka1*(x(Siah)+x(Reprimo)-ka2)}};$$

We take the derivative of this to embed it into the ODE set.

For $Rb$ = 1, the Rb controls are working correctly. In that case, low levels of the Cyclin E/CDK2 complex (ECDK2) will arrest the cell cycle, independently of SIAH and Reprimo levels. On the other hand, high levels of ECDK2 mean that the cell can pass through the G1-S transition, but SIAH or Reprimo might still stop it. We model this as a convex combination for the arrest signal:

*Case Rb = 1:*

$$oArrestsignal = \lambda_{\text{low}} * 1 + (1 - \lambda_{low}) \ \frac{1}{1+e^{ka1*(x(Siah)+x(Reprimo)-ka2)}}$$

where $\lambda_{low} = 1 - \left(\frac{ECDK2}{ECDK2_{max}}\right)$, where $ECDK2_{max}$ is the maximum level that $ECDK2$ can attain.

We differentiate this as above.

The final classification (into one of four states: 1, 2, 3, or 4) for the ground truth simulation uses the following rules, based on the levels at the end of the simulation:

| | |
|---|---|
| If Apoptosis ≥ 0.8: | 1 (apoptosis) |
| Else | |
| If FIXED > 0.9 and ARREST < .5: | 2 (repaired and cycling) |
| Else | |
| If FIXED ≤ 0.9 and ARREST < .5: | 3 (not repaired, and cycling, i.e. mitotic catastrophe) |
| Else | 4 (quiescence) |

For details on mutations and parameter changes used for ground truth dataset and the kernel dataset, see the MATLAB input files.

*Flowering model*

The flowering model is taken directly from Valentim et al. (2015)[8]. The outcome that we build a prediction model

for is flowering time, which, as in the original paper, is taken to be the time at which the protein AP1 exceeds a given threshold. The ODE model is simulated using MATLAB. The flowering model represents an isolated genetic circuit in multi-cellular eukaryote, and therefore as a model is a distant cousin—but a relevant one—to the vastly complicated genetic circuitry of human cancer cells.

*Boolean cancer model*

The Boolean cancer model is taken from Cohen et al. (2015)[9]. We converted their GinSIM model into BoolNet format, which is a package in R. The authors provide an original model as well as a modular reduction. In the SimKern simulation we use the modular reduction, which represents a limited understanding of the model, further perturbed by uncertainties of how to map the feature data into this reduced model. The model output for both the ground truth dataset and the SimKern runs are based on the steady state vectors found by simulating the network for the given initial conditions. Let $ss(n)$ denote the steady state value for node $n$. If the steady state is a fixed steady state, $ss(n)$ will be a single value, either 0 or 1. If the steady state is a cycle, then $ss(n)$ will be a binary vector of length equal to the cycle length. For the classification, we rely on two compartments in particular: $n$ = Apoptosis and $n$ = Metastasis. We classify the outputs into three categories using the following logic.

| | |
|---|---|
| If all(ss(Apoptosis)) = 1: | 1 (apoptosis) |
| Else | |
| If all(ss(Metastasis)) = 1: | 2 (metastasis) |
| Else | 3 (other) |

For details about the meaning of this model we refer the readers to the original publication[9]. In the present work, it is sufficient to view this model as an instance of a discrete complex system.

*Network flow optimization model*

We wrote a random network generation routine in MATLAB which generates a layer-wise directed graph. The user specifies the number of nodes for each layer and probabilities for adding a connecting arc between the nodes of two layers. We also add arcs between non-adjacent layers with a small probability. We ran this routine once to create a single network for all the samples in the dataset, shown in Figure S2. We generate random numbers for the cost for these arcs. This represents the base network from which all the samples of SIM0 are built. Unique samples are created by varying the weights of 12 of the 80 arcs, the bold arcs in Figure S2. The outcome of the simulation is a classification, 1, 2, or 3, representing which of the last three arcs the optimal flow passes through (linear network flow optimization theory guarantees that there exists an optimal solution with all the flow through one of the exit arcs, and that such a solution will be returned by simplex-based methods[10]).

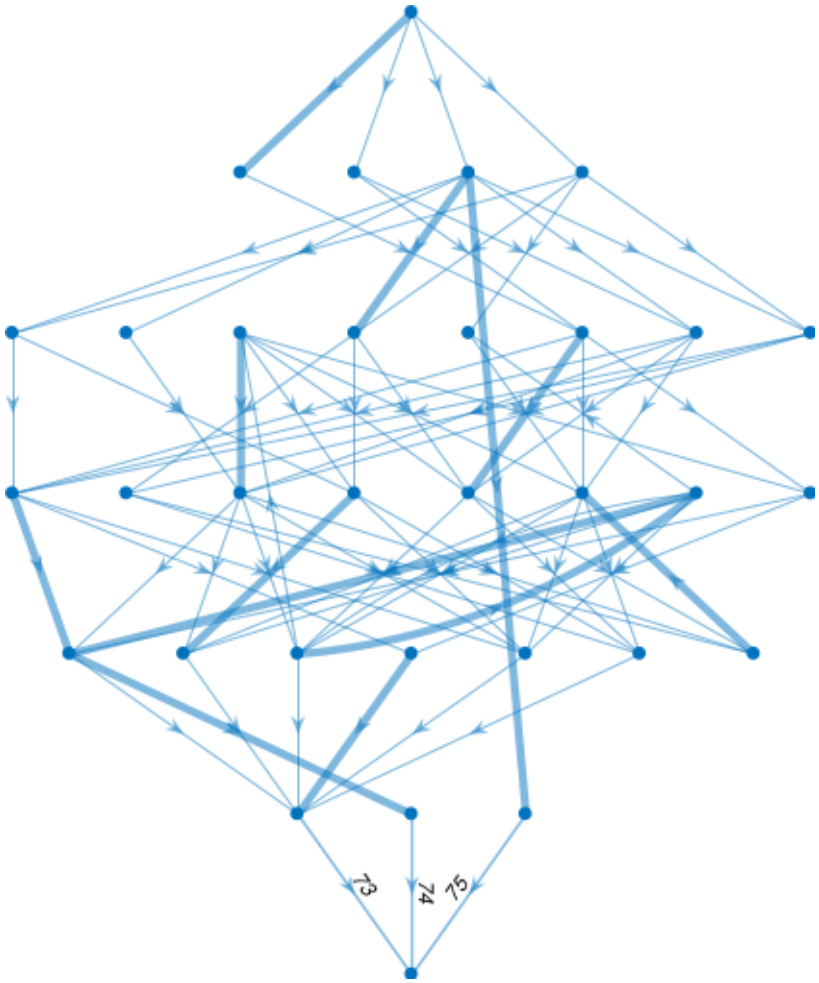



**Figure S4.** Network flow directed graph. The bold lines are the arcs with variable costs in the ground truth simulation. The unit flow that enters the network at the uppermost node will exit through one of the labeled arcs at the bottom, which creates a classification problem.

We run two versions of SIM1, a *less noisy* model (with fewer perturbed, less noisy arc costs) and a *noisier* model (with larger number and higher magnitude of perturbed arc costs). For the *less noisy* models we assume the arc costs of the 12 SIM0 variable arcs are not known with certainty: they are scaled by a uniformly distributed random variable between 0.1 and 1.9. We also perturb every arc in the second layer by a uniform variable from 0.5 to 1.5. For the *noisier* model we additionally perturb the arc costs of the third layer (uniform 0.5 to 1.5) as well as a large perturbation, uniform between 9 and 10, of the third arc, which otherwise always takes the flow because of its otherwise low arc cost (see Figure S4).

## Supplementary results

The flowering model, Figure S5, displays a typical "good kernel" result where the SimKern methods dominate the no-prior-knowledge methods throughout, but especially for small training set sizes. Similarity based NN is competitive with the more sophisticated similarity SVM and RF, but exhibits slightly more variance. The success of the SimKern methods indicates that the space induced by the similarity kernel is well behaved and the classes are easily separable with this kernel.

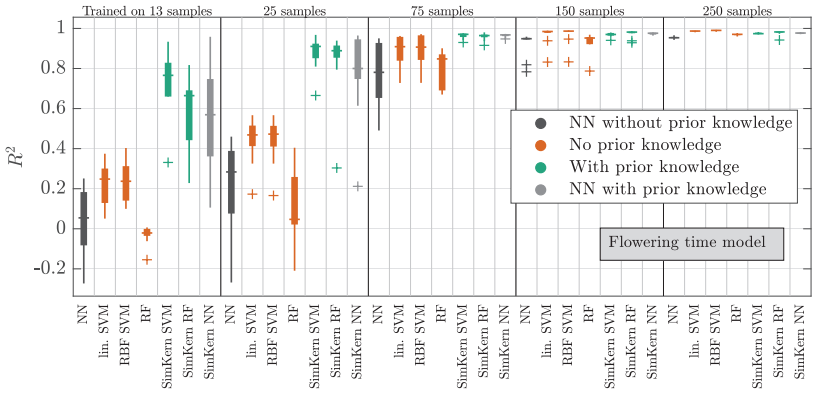



**Figure S5.** Machine learning results for the flowering model. NN = nearest neighbor, RF = random forest, SVM = support vector machine, RBF = radial basis function. $R^2$ is the coefficient of determination.

With the network flow model, we demonstrate the obvious but important result that if the SimKern simulation is farther from the ground truth simulation due to additional noise, the SimKern learning will be worse. The kernel based on a *less noisy* SimKern simulation, Figure S6, displays dominance throughout whereas the kernel based on a *noisier* SimKern simulation, Figure S7, is overtaken by the standard RF already by 18 training samples. We also used vector-based outputs from the SIM1 simulations, where the flow through every arc was used to compute similarity scores. The results were not fundamentally different so here we display results from only the scalar based SIM1 output.

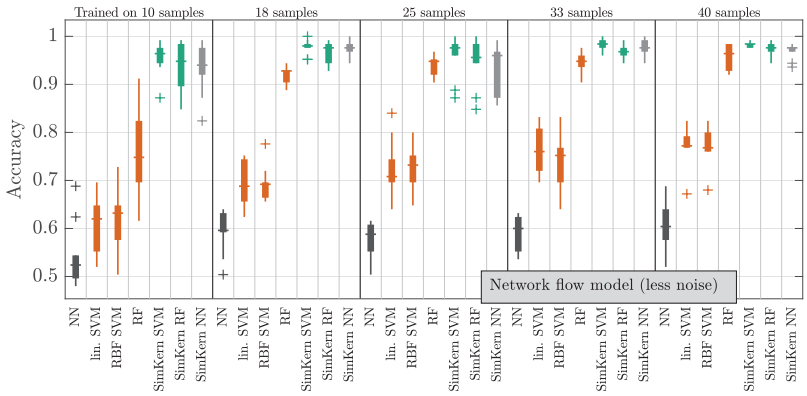



**Figure S6.** Machine learning results for the network flow optimization model for the *less noise* case. NN = nearest neighbor, RF = random forest, SVM = support vector machine, RBF = radial basis function.

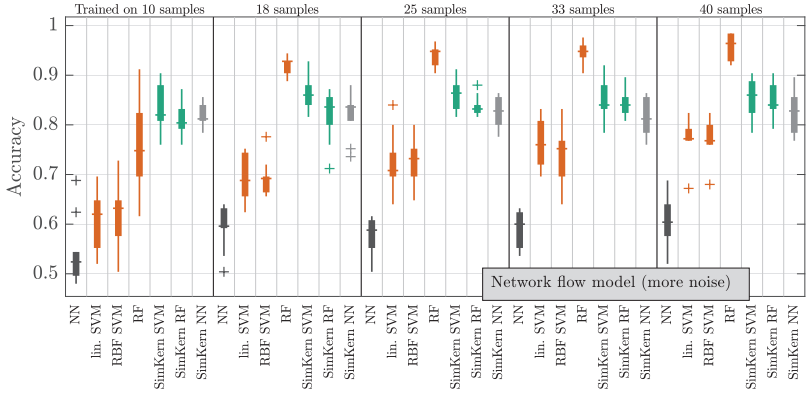



**Figure S7.** Machine learning results for the network flow optimization model for the *more noise* case. NN = nearest neighbor, RF = random forest, SVM = support vector machine, RBF = radial basis function.

The SimKern idea is effective provided that the simulations correctly judge the similarity between two samples, but the SimKern simulations need not themselves make correct predictions (in fact, the raw output of the SimKern simulations need not be the same type of output as we are trying to predict). To illustrate this, we examine the first 13 samples from the dataset for the network (lower quality) model, see Figure S8. Samples 2 and 11, which both are classified as 3s in the ground truth dataset, are given a high similarity score because they behave similarly for most of the 10 trials, even though in only one of those trials (trial 6) are they actually classified correctly.

Each model displays two nearest neighbor (NN) algorithm learning results: the default method which is Euclidean distance in feature space, and the kernelized method which uses the simulation based similarity scores for the distance computation. Consistently, the kernel based NN methods dominates over standard NN, which implies the power of a custom similarity measure. The difference between either of these NN methods and the SVMs display the power of better machine learning algorithms: rather than classifying a new sample based on which training sample it is closest to, SVMs factor in the distance to many of the training samples. In some cases (Figure 3, main document: the radiation model with the higher quality kernel, and Figure 4 main document or Figure S5: the flowering model) we see that a good similarity score is ultimately good enough and more advanced machine learning algorithms do not offer much improvement over the kernelized NN.

| Sample → | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Trial 1 outcome | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 1 | 1 | 2 |
| Trial 2 outcome | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3 | 1 | 1 | 1 | 1 | 1 |
| Trial 3 outcome | 1 | 1 | 1 | 1 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | 1 | 1 |
| Trial 4 outcome | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3 | 1 | 1 | 1 | 1 | 1 |
| Trial 5 outcome | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Trial 6 outcome | 1 | 3 | 1 | 3 | 1 | 3 | 2 | 3 | 1 | 1 | 3 | 1 | 3 |
| Trial 7 outcome | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Trial 8 outcome | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 1 |
| Trial 9 outcome | 1 | 1 | 1 | 1 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | 1 | 1 |
| Trial 10 outcome | 1 | 2 | 2 | 1 | 2 | 1 | 2 | 2 | 2 | 2 | 1 | 1 | 2 |
| Ground truth | 1 | 3 | 1 | 3 | 1 | 3 | 2 | 3 | 1 | 1 | 3 | 1 | 1 |

**Figure S8.** SIM1 results for the first 13 samples from the network (lower quality) dataset, for all ten trials and also showing in the bottom yellow row the ground truth (SIM0) result. We have highlighted samples 2 and 11. These samples are both 3s in the ground truth set, but in the $R = 10$ SimKern (SIM1) trials they get correctly classified only once. However, they are given a high similarity score since they behave the same for most of the trials. We use this to highlight the idea that it is sufficient to correctly judge sample similarity; accurate class prediction is not necessary.

As an additional way to compare machine learning results in the case of regression (the flowering model), Figure S9 plots the predicted flowering times versus the actual flowering times. With additional training samples (13 to 25), linear SVM and, even more so, SimKern SVM improve their predictions for samples with a flowering time < 6. After training on additional data, one observes a small additional downward bias in linear SVM predictions for samples with a flowering time > 10. Both algorithms, however, achieve an $R^2$ improvement by 0.19 (linear SVM) and 0.31 (SimKern SVM).

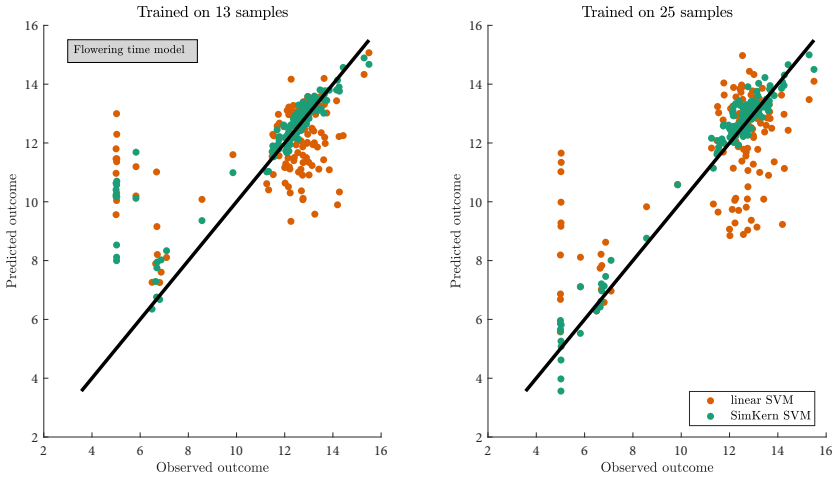



**Figure S9.** Observed and predicted test set values after training on 13 (left) and 25 (right) samples for the flowering time model. Results for linear SVM and SimKern SVM are in orange and green, respectively. Left: $R^2$ equals 0.27 and 0.66 for linear SVM and SimKern SVM, respectively. Right: $R^2$ equals 0.46 and 0.97 for linear SVM and SimKern SVM, respectively.



### *A word on the "kernel trick"*

Kernel methods are often touted in the literature as a cure-all for the problem of overly high dimensional samples: by kernelizing the data, the high dimensionality goes away. In fact, kernelizing data does not so cleanly solve this problem since there are many ways to make a kernel. Only when considering highly restricted kernel classes such as linear kernels or RBF kernels, without any feature weighting or feature selection, does the kernel trick simplify the search for a good machine learning approach. But in general, we do not know how to build a good kernel (that is, how to judge similarity between two samples in a way that is most effective for our machine learning problem). We propose herein to distill expert knowledge of a domain into simulations that use the high dimensional features, which pre-supposes quite detailed knowledge of the system. If such detailed knowledge is not available, the number of ways to turn a large feature set into a kernel is unmanageable (consider combinatoric calculations for example of selecting 200 genes out of 20000 to test all sets of 200 genes). We state this here as a word of caution: the kernel approach can be very useful but it requires obtaining a good kernel, and there is no general recipe for that.

Clearly for the SimKern approach to work, the simulations used to generate the kernel have to be "good", but unfortunately, it does not seem possible to be more quantitative than that for general cases. We explored the issue by demonstrating that as we veer away from high quality simulations, the machine learning using the custom kernel does worse (see e.g. Figures S6 and S7), but it will always be a data- and problem-specific analysis to see if a proposed kernel is useful.

## References


1. Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, **2**:27:1-27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
2. Asa Ben-Hur and Jason Weston. A users guide to support vector machines. *Data mining techniques for the life sciences*, pages 223-239, 2010.
3. Leo Breiman. Random forests. *Machine learning*, **45**(1):5-32, 2001. 25
4. Roxane Duroux and Erwan Scornet. Impact of subsampling and pruning on random forests. *arXiv preprint arXiv:1603.04261*, 2016.
5. Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, **20**(3):273-297, 1995.
6. Saket Sathe and Charu C Aggarwal. Similarity forests. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 395-403. ACM, 2017.
7. Ján Eliaš, Luna Dimitrio, Jean Clairambault, and Roberto Natalini. The p53 protein and its molecular network: modelling a missing link between dna damage and cell fate. *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics*, **1844**(1):232-247, 2014.
8. Felipe Leal Valentim, Simon van Mourik, David Posé, Min C Kim, Markus Schmid, Roeland CHJ van Ham, Marco Busscher, Gabino F Sanchez-Perez, Jaap Molenaar, Gerco C Angenent, et al. A quantitative and dynamic model of the Arabidopsis flowering time gene regulatory network. *PloS one*, **10**(2):e0116973, 2015.
9. David PA Cohen, Loredana Martignetti, Sylvie Robine, Emmanuel Barillot, Andrei Zinovyev, and Laurence Calzone. Mathematical modelling of molecular pathways enabling tumour cell invasion and migration. *PLoS computational biology*, **11**(11):e1004571, 2015.
10. D. Bertsimas and J. Tsitsiklis. *Introduction to linear optimization*. Athena Scientific, 1997.

# Chapter 6

## Discussion

The previous chapters have addressed innovations and findings to foster machine learning for radiotherapy and medical research:

- distributed data infrastructures (chapters 2 & 3),
- how to train prediction models in such distributed setting (chapters 2 & 3),
- a systematic comparison of machine learning algorithms on clinical data (chapter 4),
- a new method to combine simulation studies and kernelized machine learning algorithms (chapter 5).

In this chapter, I will discuss challenges to successfully establishing distributed data infrastructures and to the use of machine learning in radiotherapy research.

## Distributed data infrastructures

A major aspect of the distributed learning projects described in chapters 2 & 3—besides the actual learning of prediction models—is creating access to patient databases distributed over hospitals, thereby increasing the amount of patient data available for machine learning. The advantages of distributed data storage and access for research have been discussed in chapter 2. However, this data infrastructure faces multiple challenges. I will elaborate on the two most prominent challenges: infrastructure sustainability and acceptance by the research community.



### *Infrastructure sustainability*

Sustainability of the distributed data infrastructure introduced with euroCAT (chapter 2) and extended in follow up projects such as The Personal Health Train (chapter 3) depends on proper technical implementation and continuous maintenance.

Firstly, sustainable technical implementation of the distributed learning infrastructure does not coincide with the goals of funding agencies and professional goals of researchers. Distributed learning research projects are funded by public research grants which are limited to a few years and the implementation is carried out by, among others, junior researchers with short term goals. The success of research projects and the performance of junior researchers is evaluated by the quantity and quality of published manuscripts within the funded period. Manuscripts highlight individual results and are not suitable for assessing the technical quality of infrastructure implementations. Therefore, working towards publishable findings will take precedence over implementation quality.

Furthermore, software is developed by researchers-in-training rather than professional software developers. The development process may, therefore, also be their first large implementation experience. As a consequence, the resulting product is more similar to a prototype than a tool for universal use.

Secondly, continuity in technical maintenance and project management are threatened by lacking financial support in common research financing schemes and researcher turnover. After completing research for a given grant and the corresponding funding ends, infrastructure maintenance drains hardware and human resources of local hospitals. Unless a follow-up grant is acquired or local researchers are motivated to raise internal support for the maintenance of the local infrastructure components, the infrastructure endpoint is removed and future access to data is wasted. These issues where factors contributing to the loss of the euroCAT infrastructure (chapter 2) after completion of the project. To reduce the risk of losing old endpoints, collaborating hospitals need to be actively involved in the research project to emphasize the infrastructure's benefits and invoke local researchers' support for the infrastructure after the research funding has been depleted.

Another risk for the continuity of the infrastructure is that academic research projects and the corresponding knowledge are centered on individual researchers. As customary in academia, PhD students quit the project after graduation and post-doctoral researchers are likely to change employers in the search of tenure track positions. The resulting employee turnover—combined with the difficulties to organize extensive knowledge transfer in complex research projects—hinders continuity in academic infrastructure projects.

For the distributed data infrastructure to succeed in the long run, a permanent funding source needs to be uncovered. Permanent funding will finance resources to keep existing infrastructure and (permanently) employ qualified personnel to maintain and develop software for general use. The first step towards this goal has already been taken when the Health-RI initiative (an umbrella initiative with structural funding, for projects such as euroCAT and The Personal Health Train) was placed on the KNAW agenda for large-scale research facilities[1].

*Infrastructure acceptance*

The long-term vision of distributed data infrastructures is to provide access to routine clinical care data in (all) radiotherapy institutes worldwide to boost machine learning research in radiotherapy and provide insights that are useful in clinical practice. The path towards this utopia is long and requires continuous support by researchers, administrators, and funding bodies. Acceptance by researchers is crucial, especially by those who are only users and not also developers of the infrastructure, because only proof of a wide user base will convince governmental funding agencies to finance an (inter)national roll-out of the infrastructure.

Creating acceptance among researchers is a weak point of this distributed data infrastructure: the user experience in the distributed data analysis process is fundamentally different from the current centralized data analysis process known to researchers. The main difference is that the researcher cannot view the data while working on the analysis. Instead, they have to rely on summary statistics and model coefficients returned by each hospital. Furthermore, convenient data pre-processing and statistical functions present in established software packages are not readily available and need to be implemented for the distributed setting. Both aspects cause inconvenience to the researcher and likely result in preference for the existing centralized data analysis. It is possible to alleviate the inconvenience by developing software packages for distributed data analysis with a wide range of commonly used functions. However, improving the user experience will require time and resources that are predominantly spent on generating publishable results (see above). Furthermore, a strategic decision would need to be made on which software solution to support. Currently, even within our research group, which is developing the distributed data infrastructure, there is no consensus on the distributed learning framework (Varian Learning Connector or an open-source alternative) or on the programming language used to process data and train machine learning algorithms (MATLAB, R, Python, or Java). In their analysis of large historical infrastructure projects (e.g., electricity networks, internet), Wittenburg and Strawn (2018)[2] name this phase 'creolization': exploring various possible solutions before converging to universal standards which then enables large scale exploitation of the infrastructure. In our case, we need to shorten or postpone this exploration of possible solutions in exchange for quickly developing a preliminary but stable and user-friendly infrastructure to increase confidence with funding agencies. Achieving user acceptance and securing institutional support for the distributed

data infrastructure are more valuable for proliferating machine learning in radiotherapy than academic squabbles over optimal infrastructure designs.

One particular design choice in the distributed data infrastructure bears the risk to hamper acceptance among researchers: in each hospital, patient features are mapped to standardized terms in an ontology (e.g., the ROO ontology[3]) and data tables are translated into triples. Patient data is then queried using SPARQL[4]. The advantages of semantic web technology lie in the urgently needed feature standardization and reasoning capabilities. While adopting this versatile technology may be beneficial in the long run, it poses multiple difficulties that threaten acceptance of the distributed data infrastructure. Firstly, users changing from centralized to distributed data analysis will not only have to accept the lack of visual access to the data and limited analysis functionality (see above), but they will also have to learn and use the, arguably complex, SPARQL. Secondly, when researchers wish to investigate a novel feature collected at some hospital, it will first need to be defined in the corresponding ontology before it can be mapped to triples and eventually be queried by the user. The same problem occurs for existing but less common features that were not yet defined in ontologies. Ontology design is a complex task in itself and will probably reside with specialized individuals which means that progress in the user's research will depend on the availability of key developers/maintainers of the distributed data infrastructure. This process may become lengthy and obscure to a user who initially 'just wanted to query a column'. There are alternatives to semantic web technology, e.g., storing features in relational databases defined by a (local or global) data dictionary accessed using SQL. Settling with a less flexible but established method would offer the advantage that it will not overburden new users of the distributed data infrastructure. Nonetheless, it is difficult to choose between introducing a future-proof technology and conventional methods for the sake of acceptance. The former will, however, require substantial user training and support so that the distributed data infrastructure becomes a success.

In conclusion, unless significant effort and resources are put into simplifying the user experience, the aforementioned drawbacks—lack of visual access to data, limited functionality, parallel introduction of semantic web technology—may severely hamper user acceptance and consequently adoption of distributed data infrastructures in radiation oncology.

## Machine learning methodology and publications

Machine learning has received great attention in the research community but we are yet to see a machine learning application in oncology that changes clinical practice. Commercial high-profile projects have been met with anticipation, but the case of IBM Watson for Oncology exhibits a discrepancy between advertised benefit and clinical reality[5]: instead of AI-generated treatment recommendations, it is said to provide advice trained by medical doctors.

Barring clinical implementation, already at the level of scientific publications, the undifferentiated use of machine learning methods and overstating findings will disappoint and eventually tire the audience, publicly flawing the concept of machine learning in radiotherapy.

The hype surrounding machine learning should not tempt researchers to overestimate capabilities of machine learning algorithms and ignore scientific standards. Specifically, not all prediction tasks in radiotherapy become solvable by using more complex machine learning algorithms or by sequentially trying many different algorithms.

In chapter 4 we have shown based on 12 radiotherapy datasets that classifiers perform differently across datasets and one can pick a better classifier for a certain dataset—but we also see that the more complex algorithm is not always the best choice, e.g., a single hidden layer neural network performs worse than a simpler penalized logistic regression.

For an algorithm to correctly predict outcomes of a patient,

- the necessary information needs to be present in the patient's data,
- the algorithm needs to have been trained on sufficient patient cases to correctly separate useful and unnecessary information,
- the algorithm needs to be complex enough to correctly model the information.

When a simple algorithm fails patient predictions, the problem might lie in any of the three conditions. As machine learning research in radiotherapy is chronically short on patient data, the easiest solution is to replace the algorithm and redo the training and validation. Adjusting the algorithm (and therefore the underlying model) to the classification problem is a legitimate and recommendable step in the modelling process. If done correctly, it allows assessing the added benefit of a change in algorithm complexity. Extending the analysis to multiple algorithms, however, also bears a risk to flaw the analysis: training and validating different algorithms on a fixed pair of training and validation datasets is a clear case of multiple hypothesis testing and needs to be taken into account.

The now available array of machine learning algorithms therefore adds yet another way to (unintentionally) report overly optimistic modelling results. Increasing the algorithm's complexity can conveniently be justified as 'using innovative methods' in the current machine learning buzz. (Similarly, decreasing algorithm complexity could be defended as 'conservative modelling' following Occam's razor.)

For that reason, researchers need to become aware of these pitfalls. When deliberating the use of more complex machine learning algorithms, they should ask themselves whether the lack of algorithm complexity is the most likely cause of poor predictions or whether it is more probable that they train their algorithm on unrealistically few patient cases and/or uninformative features.

If indeed too few patient cases or uninformative features are used and there is no reasonable way to get access to more data or better features, pure machine learning approaches might not be able to solve the prediction tasks. A way to possibly improve models is to add domain knowledge: e.g., expert-guided feature selection[6,7], expert-built decision trees or expert-validated prediction models[8]. Expert knowledge is, however, not without flaws: when asking

experts to predict treatment outcomes for patients, they perform worse than fully data-driven machine learning algorithms[9]. Therefore, it needs to be assessed for which step in the modelling process expert knowledge can be of added value.
Related to the goal of including domain knowledge during the machine learning modelling process, we have described a new general methodology (chapter 5) that allows incorporating (biological) simulations in kernelized machine learning algorithms. While this will certainly not solve the problem of limited data and uninformative features, it is an avenue to combine the vast, decade-old expertise on biological simulations with the now popular machine learning algorithms.

To preclude a 'replication crisis' as observed in other fields[10,11], more rigorous standards need to be adopted for machine learning research in radiotherapy. It is good to see that standards like the TRIPOD statement[12], which defines steps to assess prediction model performance, are already used for machine learning research in radiotherapy. For radiomics studies, we proposed the radiomics quality score (RQS)[13] to assess manuscript quality. To further avoid mistakes in the statistical analysis of machine learning algorithms and reduce pressure on researchers to present only positive results, 'pre-registration' of studies[14,15] should be promoted: a study design is peer-reviewed and accepted before data is collected and analyzed, the eventual result will be accepted regardless of the conclusions. The counter argument that pre-registration would stifle valuable exploratory analyses is invalid as pre-registration will still allow reporting additional results in the final report although labelled as 'exploratory analysis' as argued in Chambers et al. (2013)[14].
Radiotherapy and medical physics journals would do machine learning researchers a service if they introduced and promoted study pre-registration.



**Future prospects**

In the optimal case, distributed data infrastructures will receive nationwide support by the Dutch government to collect and standardize patient features as exemplified in The Personal Health Train project (chapter 3). Recent developments show that there is indeed public interest in the concept of distributed data infrastructures. Distributed data infrastructure collaborations are being initiated between, e.g.,

- the Dutch and Taiwanese cancer registry[16],
- Statistics Netherlands (CBS) and the Maastricht Study on Diabetes,
- health insurers (Vektis) and government agencies (NZA),
- Dutch proton therapy centers (PROTRAIT) and Limburg health care providers (LIME).

Likely, the nine Dutch university medical centers, of which four participate in the duCAT project and one is aspiring to join follow-up projects, will take a leading role to extend the distributed data infrastructure and develop distributed machine learning models through the aforementioned Health-RI initiative. Regional hospitals will act as data providers and users of the resulting machine learning models. Until nationwide adoption is realized, academic projects will need to continue highlighting the use of this infrastructure and the benefits for radiotherapy. As stressed earlier, infrastructure sustainability and usability need significant attention.

The use of machine learning applications in radiotherapy clinics will be inevitable. Only the extent to which they will assist medical professionals in the radiotherapy process will be affected by how machine learning research will be conducted and communicated in the near future.Machine learning applications to help with repetitive, time consuming but somewhat

easier tasks such as metastatic node detection[17] or organ and tumor delineation[18] will soon be adopted in clinical care. Delegating complex decision making processes, such as treatment selection between chemoradiotherapy versus radiotherapy only, to machine learning models will require unwavering trust in and understanding of these algorithms by medical professionals. Obscure machine learning methodology and a publication culture with non-replicable results will undermine the necessary trust.
Once a machine learning application has been developed, prospectively validated in clinical trials, and approved for medical use, it will enter radiotherapy clinics like any other technology: the algorithm will need to be commissioned in each clinic using local patient data. Therefore, machine learning models will also become available for smaller clinics without their own active research groups.

## Conclusion

There is still a long way ahead of us for machine learning algorithms to change radiotherapy practice. The enthusiasm in the scientific community to develop practice-changing machine learning applications is high. We need to use this momentum while it lasts to lay the groundwork for a sustainable future of machine learning in radiotherapy: an established distributed data infrastructure with first simple but robustly validated machine learning applications. In this thesis, we have tried to contribute to both goals with data infrastructure prototypes (chapters 2 & 3), analyses of machine learning algorithms (chapter 4), and a new machine learning methodology (chapter 5). In this way, once the novelty and excitement have ebbed away, the impetus will not have been wasted on short-term gains, but the foundations will have been laid out and the work can continue.

## References


1. Thirteen selected facilities and three honourable mentions — KNAW. Available at: https://www.knaw.nl/en/advisory-work/thirteen-facilities. (Accessed: 19th October 2018)
2. Peter Wittenburg, G. S. Common Patterns in Revolutionary Infrastructures and Data. (2018). doi:10.23728/b2share.4e8ac36c0dd343da81fd9e83e72805a0
3. Traverso, A., Soest, J. van, Wee, L. & Dekker, A. The radiation oncology ontology (ROO): Publishing linked data in radiation oncology using semantic web and ontology techniques. *Medical Physics* (accepted). doi:10.1002/mp.12879
4. Prud'Hommeaux, E. & Seaborne, A. SPARQL query language for RDF. *W3C recommendation* **15,** (2008).
5. IBM pitched Watson as a revolution in cancer care. It's nowhere close. Available at: https://www.statnews.com/2017/09/05/watson-ibm-cancer/. (Accessed: 19th October 2018)
6. Deist, T. M. *et al.* OC-0139: Expert knowledge vs. data-driven algorithms: Bayesian prediction models for post-radiotherapy dyspnea. *Radiotherapy and Oncology* **119,** S62–S63 (2016).
7. Deist, T. M. *et al.* 60 - Expert knowledge and data-driven Bayesian Networks to predict post-RT dyspnea and 2-year survival. *Radiotherapy and Oncology* **118,** S29–S30 (2016).
8. MEDIFORESTS. Available at: http://www.mediforest.com/login. (Accessed: 19th October 2018)
9. Oberije, C. *et al.* A prospective study comparing the predictions of doctors versus models for treatment outcome of lung cancer patients: A step toward individualized care and shared decision making. *Radiotherapy and Oncology* **112,** 37–43 (2014).
10. Collaboration, O. S. Estimating the reproducibility of psychological science. *Science* **349,** aac4716 (2015).
11. Gilbert, D. T., King, G., Pettigrew, S. & Wilson, T. D. Comment on "Estimating the reproducibility of psychological science". *Science* **351,** 1037–1037 (2016).
12. Collins, G. S., Reitsma, J. B., Altman, D. G. & Moons, K. G. Transparent reporting of a multivariable prediction model for individual prognosis or diagnosis (TRIPOD): the TRIPOD Statement. *BMC Medicine* **13,** 1 (2015).
13. Radiomics: the bridge between medical imaging and personalized medicine | Nature Reviews Clinical Oncology. Available at: https://www.nature.com/articles/nrclinonc.2017.141. (Accessed: 23rd September 2018)
14. Chambers, C., Munafo, M. & signatories, more than 80. Trust in science would be improved by study pre-registration. *The Guardian* (2013).
15. PainDec. 15, E., 2015 & Pm, 5:30. Register your study as a new publication option. *Science | AAAS* (2015). Available at: https://www.sciencemag.org/careers/2015/12/register-your-study-new-publication-option. (Accessed: 19th October 2018)
16. Geleijnse, G. Distributedlearning.ai: Towards a Distributed Learning Network for Cancer Registries. *ENCR Scientific Meeting* (2018).
17. Ehteshami, B. B. *et al.* Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer. *JAMA* **318,** 2199–2210 (2017).
18. Lustberg, T. *et al.* Clinical evaluation of atlas and deep learning based automatic contouring for lung cancer. *Radiotherapy and Oncology* **126,** 312–317 (2018).

## Conflict of interest

While working on his PhD thesis, Timo Deist held a part-time employment with ptTheragnostic B.V. as scientist to develop mtDNA-based biomarkers.

# Appendices

# Appendix I

## Summary

This thesis discusses machine learning methods to analyze patient data in radiation oncology. Due to the increasing availability of computing power, digitalization of medical records, and the success of machine learning in other fields, increasingly sophisticated data analysis procedures are sought after also in the medical sciences. In radiation oncology, such machine learning methods might improve the prediction of radiotherapy outcomes for individual patients, which could aid physicians and patients in selecting suitable treatments. Current research efforts are dedicated to attaining a thorough understanding of the existing machine learning methods and how they can be used for treatment outcome prediction.

Furthermore, developing reliable machine learning applications requires access to large amounts of patient data. Patient data is stored by each healthcare provider and collaboration between these healthcare providers is needed to reach sufficient patient data volumes. Regulatory barriers to protect patient-privacy complicate sharing patient data across healthcare providers and therefore hamper the implementation of machine learning applications in clinical practice. Technological solutions are needed to allow machine learning across healthcare providers while meeting privacy regulations. The existing concept of *distributed learning* might pose a solution, i.e. training machine learning algorithms on patient data stored at distinct healthcare providers without patient data being exchanged.

The studies presented in this thesis form two parts:

- the development of a distributed learning infrastructure to facilitate privacy-preserving machine learning studies across healthcare providers
- the analysis of existing machine learning methods in the context of radiotherapy outcome prediction and the development of a new machine learning method.

With a small proof-of-concept study in chapter 2, the distributed learning infrastructure was described and it was shown that distributed learning across radiotherapy institutes is possible. Support vector machine models to predict post-radiotherapy dyspnea (grade 2 or higher) were trained on lung cancer patient data from five radiotherapy clinics located in three countries (Belgium, Germany, The Netherlands).

To demonstrate the infrastructure's scalability, another study in chapter 3 applied this infrastructure in eight healthcare providers across five countries (China, England, Italy, The Netherlands, Wales) to train and validate a logistic regression for predicting post-treatment two-year survival based on tumor staging data of more than 20 000 non-small cell lung cancer patients. This study was executed in four months demonstrating the distributed learning infrastructure's potential for fast-paced machine learning studies.

In a study of existing machine learning methods for (chemo)radiotherapy outcome prediction (chapter 4), the discriminative performance of six machine learning algorithms (decision tree, random forest, neural network, support vector machine, elastic net logistic regression, LogitBoost) was compared and ranked on twelve patient data sets. Random forest and elastic net logistic regression showed a small increase in discriminative performance. These findings might guide researchers in selecting appropriate machine learning methods for future studies.

Finally, a new kernelized machine learning approach was presented in chapter 5, which allows combining simulation models and machine learning methods. Both simulations and machine learning methods allow inferring predictions: simulation models use prior knowledge gained by (experimental) analysis of a system while machine learning methods derive predictions from data with statistical means. The presented results from four synthetic scenarios indicated that merging simulation models and machine learning methods might pose an advantage in scenarios where insufficient data is available to train standard machine learning algorithms. Chapter 6 discussed challenges and future prospects for distributed learning infrastructures and the use of machine learning methods in radiation oncology. Challenges for distributed learning addressed in this chapter are infrastructure sustainability and its acceptance by users. Furthermore, the risks of misusing machine learning methods and overstating results, and how reporting standards and pre-publication registration of studies can mitigate negative consequences are considered.

# Appendix II

## Valorization

For distributed data infrastructures and machine learning algorithms to last in radiotherapy, they need to add value to society and/or be capitalized in the private sector.
Distributed data infrastructures may form integral parts of future radiotherapy clinics for the benefit of society and each individual patient. Therefore, (inter)national governments may decide to mandate it for healthcare providers.
It is in the interest of the public to

- foster access to patient data for medical research,
- control the quality of care by healthcare providers,
- ensure patient privacy and control over the data.

Distributed data infrastructures support these three aspects. It is possible to apply machine learning algorithms or other kinds of data analysis processes via the infrastructure. Similarly, it is possible to compare treatments across clinics using statistical analyses. Most importantly, the data always remains at the institute where the data was generated (i.e. where the patient was treated) and the external analyst does not have direct access to the data. Standardization of medical data on an (inter)national level may pose the biggest challenge which is, however, inevitable regardless whether centralized or distributed data infrastructures are used. Therefore, it will be in the interest of public health to support or even prescribe participation in distributed data infrastructures for radiotherapy clinics.

Distributed data infrastructures also have commercial applications: medical research companies require access to patient data for pharmaceutical, device, or software development. A distributed data infrastructure would provide patients a platform to sell restricted access to their data while maintaining control and ensuring anonymity.
The Varian Learning Portal[1] (chapters 2 & 3) is evidence that the private sector sees promise in distributed data infrastructures for radiotherapy: it is free to use for radiotherapy clinics to learn prediction models using data from participating institutes but Varian has the first right of refusal for commercialization of the resulting models.

Machine learning algorithms for radiotherapy (and other medical applications) have applications with substantial benefit to society and clear commercialization prospects.
Machine learning models have the potential to assist medical professionals in repetitive tasks and complex decision-making processes:

- organ/tumor delineation[2],
- treatment planning quality control[3],
- decision support systems for treatment selection[4].

Most notably, the guidelines to select patients for proton therapy in the Netherlands prescribes a model-based decision process in which patient cases with certain diagnoses will be evaluated using (machine learning) models[5].
Reducing the time spent on these tasks saves resources and thus decreases public healthcare spending. Assisting medical professionals in making better decisions improves healthcare outcomes. Therefore, society will benefit if properly tested machine learning models become part of the radiotherapy process.

The development of machine learning models for medical applications is a long and expensive process in a heavily regulated industry but with a large global market. Individual hospitals, whose only focus is to treat their patients, will hesitate to pursue this enterprise given the high costs but private investors and multinational companies have the means and interest to finance the development at the prospect of high future payoffs. IBM Watson for Oncology[6] is an example for a large multinational corporation to develop decision support systems for oncology but investors also finance small businesses: two examples originating from Maastricht University/MAASTRO clinic are ptTheragnostic B.V.[7], which is working on decision support for proton radiotherapy, and Oncoradiomics SA[8], which is working on image-based biomarkers for radiotherapy.

In conclusion, distributed data infrastructures and machine learning algorithms for radiotherapy have clear valorization prospects both for the benefit of society and commercialization.

## References


1. VLP. Available at: https://www.varianlearningportal.com/vlp/Loading. (Accessed: 19th October 2018)
2. Lustberg, T. *et al.* Clinical evaluation of atlas and deep learning based automatic contouring for lung cancer. *Radiotherapy and Oncology* **126,** 312–317 (2018).
3. Tomori, S. *et al.* A deep learning-based prediction model for gamma evaluation in patient-specific quality assurance. *Medical Physics* **45,** 4055–4065 (2018).
4. Cheng, Q. *et al.* Development and evaluation of an online three-level proton vs photon decision support prototype for head and neck cancer – Comparison of dose, toxicity and cost-effectiveness. *Radiotherapy and Oncology* **118,** 281–285 (2016).
5. Langendijk, J. A. *et al.* Selection of patients for radiotherapy with protons aiming at reduction of side effects: The model-based approach. *Radiotherapy and Oncology* **107,** 267–273 (2013).
6. IBM Watson for Oncology - Overview - United States. (2018). Available at: https://www.ibm.com/us-en/marketplace/ibm-watson-for-oncology. (Accessed: 19th October 2018)
7. ptTheragnostic | Enabling the right treatment for the right patient. Available at: http://www.pttheragnostic.com/. (Accessed: 19th October 2018)
8. Oncoradiomics - A I - Radiomics Software - Clinical & Research. Available at: https://www.oncoradiomics.com/. (Accessed: 19th October 2018)

# Appendix III

## Acknowledgments

I would like to thank my three supervisors:
Philippe Lambin for showing me the ropes in academia and providing me with so many opportunities during my time as a PhD student.
Andre Dekker for his patient mentorship, valuable advice, his perpetual readiness to help solving problems whenever and wherever, and for reminding his students that the PhD studies should sometimes also be fun.
Arthur Jochems for the companionship, collaboration, and advice in the trenches and at the troughs.

I would also like to thank David Craft for inviting me over to the MGH for three months: the stay in Boston marked a highlight of my PhD studies, filled with ever entertaining, challenging, and sometimes marathon-length discussions on the SimKern project.

There are many people at MAASTRO clinic and the UM without whom the past four years would only have been half as enjoyable. Therefore, I would like to thank all vlaai-eaters, lunch (non-)walkers, D-lab, KE group, and research room members for their camaraderie.

Special thanks go to Frank: we formed a great team for the last two years! We perfectly complemented each other in terms of expertise but also dangerously reinforced our shared and at times futile attention for detail.

I am very grateful to my parents. They supported me in pursuing my interests, and encouraged me to learn and understand the world around me since I can remember. Wherever I wanted to go, they made it possible. Whenever I need support, they give it.

Most of all, I am grateful to Claire. You supported me at every minute of this PhD. You have always been lenient and supportive when I was working in the evening, during weekends, or on holidays. For more than four years, you listened to daily status updates almost exclusively about things that didn't work. Without you and your support, the past years would have been so much harder. Thank you!

# Appendix IV

## Curriculum vitae

Timo was born on 17 February 1989 in Essen (Germany). In 2008, he completed his high school in Essen and started his bachelor's studies in econometrics and operations research at Tilburg University (The Netherlands). After attaining his bachelor's degree in 2012, he completed his master's degree (cum laude) in operations research and management science at Tilburg University in 2013. For both his master's and bachelor's theses, he investigated heuristic algorithms for high-dose rate prostate brachytherapy treatment planning optimization. He was awarded an Erasmus Mundus scholarship and completed a second master's degree in BioHealth computing at the University of Barcelona (Spain) and Université Joseph Fourier in Grenoble (France) in 2014.

For his master's thesis, he studied the estimation of ancestry coefficients using non-negative matrix factorization and spatial information. He then returned to The Netherlands to pursue his PhD research on distributed learning and prediction modelling at Maastricht University/ MAASTRO clinic under supervision of prof. dr. Philippe Lambin and prof. dr. Andre Dekker. Next to his research for his PhD thesis, he worked part-time at the spin-off company ptTheragnostic B.V. with the goal to develop biomarkers for radiation sensitivity in human mitochondrial DNA. Between October and December 2017, he visited dr. David Craft and the department of radiotherapy at the Massachusetts General Hospital/Harvard Medical School (United States) funded by travel grants of the European Society for Radiotherapy & Oncology (ESTRO), the René Vogels Stichting, and GROW (Maastricht University). In March 2019, he starts his postdoctoral research on multi-objective optimization algorithms for medical image registration at the Centrum Wiskunde & Informatica in Amsterdam (The Netherlands) under supervision of prof. dr. Peter Bosman.

# Appendix V

## List of manuscripts

## In preparation/submitted/accepted

Dankers, F.J.W.M., **Deist, T.M.**, ... & Dekker, A. Distributed validation and retraining of an acute esophagitis prediction model for locally advanced non-small cell lung cancer patients after (chemo-)radiotherapy – A Personal Health Train application. In preparation.

**Deist, T.M.**, Dankers, F.J.W.M., ... & Price, G., Lambin, P., Dekker, A. Distributed learning on 20 000+ lung cancer patients - The Personal Health Train. In preparation.

Bogowicz, M., …, **Deist, T.M.**, … & Lambin, P. Privacy-preserving distributed learning of radiomics to predict overall survival and HPV status in head and neck cancer. In preparation.

Shi, Z., Zhovannik, I., …, **Deist, T.M.**, … & Wee, L. "Distributed Radiomics" - a signature validation study using a Personal Health Train infrastructure. In preparation.

de Jong, E., Deist, T.M., ... & Lambin, P. Can quantitative radiomic features describe qualitative semantic features in NSCLC patients? In preparation.

de Jong, E., ..., Deist, T.M., ... & Lambin, P. Radiomics approach to predict skeletal muscle response to chemotherapy in stage IV NSCLC. Submitted.

**Deist, T.M.**, Patti, A., ... & Craft, D. Simulation assisted machine learning. Submitted.

**Deist, T.M.**, Dankers, F.J.W.M., ... & Lambin, P. Erratum: Machine learning algorithms for outcome prediction in (chemo)radiotherapy: an empirical comparison of classifiers. *Medical Physics*. Accepted.

## Published original research

Tucker, S. L., ..., **Deist, T.M.**, ... & Liao, Z. (2019). Validation of Effective Dose as a Better Predictor of Radiation Pneumonitis Risk than Mean Lung Dose: Secondary Analysis of a Randomized Trial. *International Journal of Radiation Oncology·Biology·Physics* 103, 2, 403-410.

**Deist, T.M.**, Dankers, F.J.W.M., ... & Lambin, P. (2018). Machine learning algorithms for outcome prediction in (chemo)radiotherapy: an empirical comparison of classifiers. *Medical Physics* 45 (7).

Jochems, A., **Deist, T.M.**, ... & Dekker, A. (2017). Developing and validating a survival prediction model for NSCLC patients through distributed learning across 3 countries. *International Journal of Radiation Oncology·Biology·Physics*, 99, 2, 344-352.

**Deist, T.M.**, Jochems, A., ... & Lambin, P. (2017). Infrastructure and distributed learning methodology for privacy-preserving multi-centric rapid learning health care: euroCAT. *Clinical and Translational Radiation Oncology*, 4, 24–31.

Jochems, A., **Deist, T.M.**, ... & Lambin, P., Dekker, A. (2016). Distributed learning: Developing a predictive model based on data from multiple hospitals without data leaving the hospital — A real life proof of concept. *Radiotherapy and Oncology*, 121(3), 459-467.

**Deist, T.M.** & Gorissen, B.L. (2016). High-dose-rate prostate brachytherapy inverse planning on dose-volume criteria by simulated annealing. *Physics in medicine and biology*, 61(3), 1155.

Caye, K., **Deist, T.M.**, ... & François, O. (2016). TESS3: fast inference of spatial population structure and genome scans for selection. *Molecular ecology resources*, 16(2), 540-548.

Damiani, A., ..., **Deist, T.M.**, ... & Valentini, V. (2015). Distributed learning to protect privacy in multi-centric clinical studies. *Conference on Artificial Intelligence in Medicine in Europe*. Springer International Publishing.

### Published reviews

van Stiphout, R., **Deist, T.M.**, ... & Lambin, P. (2018). How to Share Data and Promote a Rapid Learning Health Medicine? In: Valentini V., Schmoll HJ., van de Velde C. (eds) *Multidisciplinary Management of Rectal Cancer*. Springer, Cham.

Lambin, P., Leijenaar, R.T.H., **Deist, T.M.**, ... & Walsh, S. (2017). Radiomics: the bridge between medical imaging and personalized medicine. *Nature Reviews Clinical Oncology*, 14, 749-762.

Lambin, P., ..., **Deist, T.M.**, ... & Walsh, S. (2017). Decision support systems for personalized and participative radiation oncology. *Advanced drug delivery reviews*, 109, 131-153.

Lustberg, T., ..., **Deist, T.M.**, ... & Dekker, A. (2016). Big Data in radiation therapy: challenges and opportunities. *The British Journal of Radiology*, 90(1069), 20160689.

Lambin, P., ..., **Deist, T.M.**, ... & Walsh, S. (2015). Modern clinical research: How rapid learning health care and cohort multiple randomised clinical trials complement traditional evidence based medicine. *Acta Oncologica*, 54.9 1289-1300.

V