Computers in Biology and Medicine 108 (2019) 371–381

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Sensitivity analysis in digital pathology: Handling large number of parameters with compute expensive workflows

Jeremias Gomes[a], Willian Barreiros Jr.[a], Tahsin Kurc[c,d], Alba C.M.A. Melo[a], Jun Kong[b,e,f], Joel H. Saltz[c], George Teodoro[a,c,*]

[a] Department of Computer Science, University of Brasília, Brazil
[b] Biomedical Informatics Department, Emory University, Atlanta, USA
[c] Biomedical Informatics Department, Stony Brook University, Stony Brook, USA
[d] Scientific Data Group, Oak Ridge National Laboratory, Oak Ridge, USA
[e] Department of Biomedical Engineering, Emory-Georgia Institute of Technology, Atlanta, USA
[f] Department of Mathematics and Statistics, Georgia State University, Atlanta, USA

ARTICLE INFO

Keywords:
Whole Slide Image Analysis
Sensitivity analysis
Surrogate models
Microscopy
Survival analysis
Uncertainty propagation

ABSTRACT

Digital pathology imaging enables valuable quantitative characterizations of tissue state at the sub-cellular level. While there is a growing set of methods for analysis of whole slide tissue images, many of them are sensitive to changes in input parameters. Evaluating how analysis results are affected by variations in input parameters is important for the development of robust methods. Executing algorithm sensitivity analyses by systematically varying input parameters is an expensive task because a single evaluation run with a moderate number of tissue images may take hours or days. Our work investigates the use of Surrogate Models (SMs) along with parallel execution to speed up parameter sensitivity analysis (SA). This approach significantly reduces the SA cost, because the SM execution is inexpensive. The evaluation of several SM strategies with two image segmentation workflows demonstrates that a SA study with SMs attains results close to a SA with real application runs (mean absolute error lower than 0.022), while the SM accelerates the SA execution by 51 × . We also show that, although the number of parameters in the example workflows is high, most of the uncertainty can be associated with a few parameters. In order to identify the impact of variations in segmentation results to downstream analyses, we carried out a survival analysis with 387 Lung Squamous Cell Carcinoma cases. This analysis was repeated using 3 values for the most significant parameters identified by the SA for the two segmentation algorithms; about 600 million cell nuclei were segmented per run. The results show that significance of the survival correlations of patient groups, assessed by a logrank test, are strongly affected by the segmentation parameter changes. This indicates that sensitivity analysis is an important tool for evaluating the stability of conclusions from image analyses.

1. Introduction

Analysis of whole slide tissue images (WSIs) is an important component of biomedical imaging studies. WSIs capture salient information about disease morphology at the sub-cellular scale. Quantitative data computed from WSIs and their correlation with other sources of information, such as clinical and molecular data, can assist in understanding disease mechanisms as well as in disease grading and classification [1–7].

While there are several benefits in using WSIs, many challenges, such as accuracy and robustness of analysis workflows and high computation costs, have to be addressed. One of the methodology challenges is the fact that analysis workflows are often sensitive to input parameters. Sensitivity Analysis (SA) is defined as the process of quantifying the inherent variability observed in results from data analyses when the values of input parameters are varied. A SA study can be used in image analysis to test robustness of analysis results, to understand correlations among input parameters and analysis output, to guide parameter tuning, and to simplify an analysis workflow. A SA with a large number of WSIs, however, is a computationally very demanding process, which limits its use in practice.

In this work we target the SA of nucleus segmentation workflows. Segmentation of nuclei in WSIs is a critical step which extracts imaging features used in downstream analyses such as classification and

[*] Corresponding author. Department of Computer Science, University of Brasília, Brazil.
E-mail address: teodoro@unb.br (G. Teodoro).

https://doi.org/10.1016/j.compbiomed.2019.03.006
Received 2 December 2018; Received in revised form 28 February 2019; Accepted 7 March 2019

correlations. Fig. S1 (supplementary material) shows two example segmentation workflows used in this paper.

The first workflow uses morphological operations and a watershed method to separate overlapping nuclei [6]. The second one employs level set and mean-shift clustering strategy to segment and declump nuclei [8]. Table A1 (supplementary material) presents the parameters and possible parameter combinations for the two workflows. The large number of parameters in these workflows leads to several application runs, since it increases with the number of parameters studied. A SA can become infeasible to execute when there are many parameters and the processing of a single WSI takes hours.

We propose a new approach that aims to reduce the execution time of SA with the use of Surrogate Models (SMs) and parallel computing systems. SMs are compact and scalable analytic models that approximate or mimic the behavior of multivariate complex systems. These models can be employed in SA studies to replace the actual application and, because they have low execution times, their use can significantly reduce the overall SA time. A SM is built by fitting a model into a training data (tuples < parameters, output >). The training data is captured from application runs; that is, the analysis application is still executed using large input datasets to build a SM. We employ high performance computing to address this computational cost.

Surrogate Models have been used in optimization studies in aerospace systems, fluid dynamics simulations, chain management, computational fluid dynamics (CFD), among others [9–12]. In this paper, we adapt and evaluate SMs for sensitivity analysis of tissue image segmentation workflows. SA has also been employed without SMs in multiple biomedical problems [13–15] that include bone remodeling, electrocardiography analysis, venous pressure, etc. However, most of the previous studies deal with applications which are computationally less expensive than nucleus segmentation workflows, and have not systematically evaluated multiple SM strategies and SA methods. The contributions of our work can be summarized as follows:

- We propose and evaluate the use of several surrogate models to accelerate sensitivity analysis in microscopy image analysis.
- We execute SA studies with two WSI segmentation workflows and multiple cancer types. The results show that SMs lead to performance gains of 51 × with a small mean absolute error (MAE) of 0.022 to the SA indices.
- We show that variation in most significant parameters for the segmentation workflows affects conclusions (e.g., survival analysis) in downstream correlations [16,17] that use features computed from segmentation results.
- We develop an end-to-end framework to efficiently support SM-based SA in microscopy tissue image analysis. Our solution supports multiple SA methods, can carry out SA in large-scale imaging datasets, and implements performance optimizations to reduce execution time.

We expect that this level of improvement in the SA will allow for (i) a more routinely use of SA in microscopy image analysis; (ii) the evaluation of SA with big imaging datasets; and, (iii) the use of powerful SA methods that require a large parameter sampling.

2. Methods

The proposed approach is illustrated in Fig. 1. In order to carry out a SA study with a large dataset, a user (e.g., scientist or application developer) needs to: (i) deploy the target segmentation application into the Region Templates runtime system [18] for parallel execution, (ii) specify the parameters to be analyzed and their value ranges, and (iii) select a metric to measure changes in the output as the input parameter values are varied. Our implementation supports the Dice and Jaccard metrics [19,20], which are commonly used to evaluate differences between sets of objects. In our case, the sets of objects are reference image masks and masks computed by the segmentation application for each parameter set chosen by the SA process. The reference masks correspond to segmentation results computed using the default application parameters. In this way, we measure how segmentation output (masks) changes with respect to a fixed reference output as the parameters values are varied.

The user conducts the SA using our framework as follows. First, the actual segmentation application is executed multiple times using the target imaging dataset to generate a set of sampling/training results. This set of results is used as input in a phase that generates multiple surrogate models with different modeling strategies. These models are evaluated to select the one that represents the application with minimum error. The selected SM is used by the framework to replace calls to the application as the input parameters values are systematically varied by the SA method chosen by the user. All the SA methods supported in our framework (presented in the next section) can be used with SMs.

2.1. Methods for sensitivity analysis

The framework supports a variety of SA methods and sampling approaches: screening methods, such as Morris One-At-A-Time (MOAT) design [21] that can be used to quickly identify non-influential parameters; and methods to compute more informative importance measures such as Pearson's and Spearman's correlation coefficients [22], and the Variance-based Decomposition (VBD) [23]. These methods can be used separately in different scenarios or in coordination to perform initial investigations, for instance, to prune unimportant parameters before more detailed and expensive studies are employed.

SA may be local or global. Local SA analyzes the impact of small perturbations around a fixed parameter value (point in the domain) to quantify the impacts of those local changes to the output. Global SA, which is our target, measures the overall impact of a parameter as it is varied on the entire parameter domain to the output [24–26]. Although our focus is on global SA, the tools and techniques proposed here could be used to compute local SA, for instance, by instantiating different SA methods or by using MOAT with small perturbations with respect to a fixed parameter value of interest.

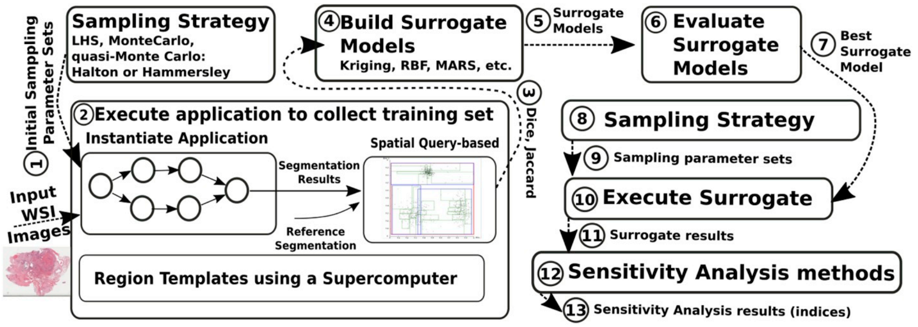

1 Initial Sampling Parameter Sets
Sampling Strategy
LHS, MonteCarlo, quasi-Monte Carlo: Halton or Hammersley
4 Build Surrogate Models
Kriging, RBF, MARS, etc.
5 Surrogate Models
6 Evaluate Surrogate Models
7 Best Surrogate Model
2 Execute application to collect training set
Instantiate Application
Segmentation Results
Spatial Query-based
Reference Segmentation
3 Dice, Jaccard
Input WSI Images
Region Templates using a Supercomputer
8 Sampling Strategy
9 Sampling parameter sets
10 Execute Surrogate
11 Surrogate results
12 Sensitivity Analysis methods
13 Sensitivity Analysis results (indices)

Fig. 1. The architecture of the SA framework. A SA method and sampling strategy are selected. The initial parameter sets to be evaluated are generated, and the application is executed using the WSIs to create a training set of results. These results are then used to build and evaluate the Surrogate Models, and the model that best represents the application is selected. Further, several calls are made to the Surrogate Model (replacing the application) until the sampling requirement of the SA method is reached and the SA results are computed.

The MOAT discretizes the input parameter space and varies a single parameter at a time to measure the difference between the application output (e.g., Dice or Jaccard) for two points in the space. In our work, the parameter value perturbation is chosen as $\Delta = p/[2(p-1)]$ [21], where p is the number of levels that a parameter assumes, to account for intervals greater than half of the parameter range and, consequently, compute for global SA. The sampling size (number of application runs) for MOAT is $n = r(k+1)$, where k is the number of parameters studied and r is typically between 5 and 15 [27].

The importance measures in our framework include Pearson's correlation coefficient (CC), partial correlation coefficient (PCC), partial rank correlation coefficient (PRCC), and Spearman's rank correlation coefficient (RCC) [22]. They compute correlations between input parameters and output or between pairs of input parameters. The partial correlation excludes effects from other input parameters. If the input parameters are independent, the simple and partial results are the same, while the ranked coefficients identify non-linear effects [22].

The Variance-based Decomposition (VBD) method [23] measures individual and non-linear relationships. It divides parameter effects in the output into main and total effects [26,28,29]. The main effect is the variation in the output due to a given parameter only, whereas the total-order effects include individual and higher-order/total-order effects. VBD demands $n(k+2)$ runs, where k is the number of parameters and n can be in the order of hundreds or thousands [23].

2.2. Surrogate models

A SM is a simplified model that creates a continuous function of a complex system using a limited amount of data [9] and emulates the behavior of said system with small execution cost [30]. A variety of surrogate models, including meta-models, response surface models, approximation models, simulation models, data-driven models [31], have been used in a wide range of application domains, such as mechanics, geology, astrophysics, and engineering [9–12].

In our case, a computational model is defined as a function f that maps an input parameter set $x_i = (x_{i_1}, x_{i_2}, \dots, x_{i_d})$ to a response value t_i, where d is the dimensionality of the problem or the number of input parameters with $x_i \in \mathbb{R}^d$.

$$f: x_i \in D(x_i, t_i) \subset \mathbb{R}^d \rightarrow f(x_i) = t_i \tag{1}$$

The surrogate model is similarly defined as a function $\hat{f}$:

$$\hat{f} = fit(Q(x_i, t_i)): \hat{f}(x_i) \approx f(x_i) = t_i \tag{2}$$

where $\hat{f}$ extrapolates from a limited training dataset Q to approximate the results produced by f. A key aspect in this context is to calibrate the surrogate based on assumptions on the $\hat{f}$ shape or behavior using a small number of runs of the application (f) for selected input configurations.

Multiple SMs have achieved good performance in other domains and, as such, were implemented and evaluated in our work: i) A Gaussian Process (GP or Kriging) that provides prediction about the variance using a construction of the covariance function examined by a pivoted Cholesky factorization [32,33]; Radial Basis Functions (RBF), which uses linear combinations based on a distance from a centroid to approximate the response function [34,35]; iii) Multivariate Adaptive Regression Splines (MARS), which is a non-parametric method that splits the domain into subregions and creates local regressions from these subregions to be combined into a continuous function for the entire domain [36]; iv)Artificial Neural Networks (NN), which are tools for modeling non-linear data that relate inputs and outputs based on the structure and functions of a biological neural network [37]; v) Support Vector Regression [38], which is a method based on supervised learning for regression analysis or data classification.

2.3. Building a surrogate model

Our framework builds a SM in a pipeline of parameter sampling, training dataset generation, and SM evaluation and selection steps.

2.3.1. Parameter Sampling

The choice of the set of parameters should be performed carefully in order to adequately represent the conditions of the application response surface. The sampling step should be sufficiently small to minimize the computational costs, and, at the same time, should still be representative and avoid bias or polarization errors. We employ full/partial random sampling (Monte Carlo or Quasi-Monte Carlo) and Latin Hypercube Sampling (LHS) [9], which assert that the sampling adequately cover the parameters space.

2.3.2. Training dataset generation (application execution)

This is conceptually the simplest step of the process. The application studied is executed for each input parameter configuration selected in the sampling stage. The application output results are measured. Tuples containing parameter sets and application response values are recorded. Training dataset generation is also the most expensive phase of the SA studies using SMs. Thus, in this phase, we employ parallel computing machines to speed up the application runs as described in Section 2.4.

2.3.3. SM Evaluation and Selection

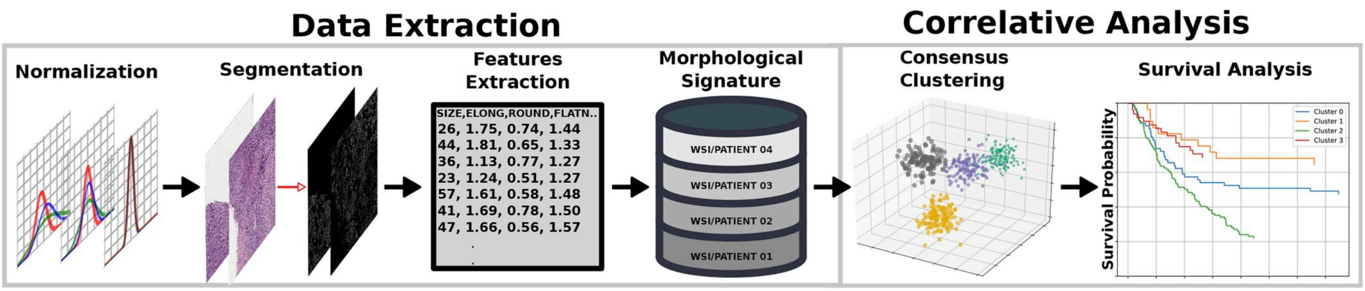
This step uses the training dataset to instantiate SMs. In our work, the application is handled as a black-box and, as a consequence, we cannot assume any information about the application that would help in selecting the appropriate SM strategy [31]. We build SMs for all strategies available and evaluate them empirically to choose the best model among those available. This evaluation is performed via cross-validation in which the training dataset is divided into two disjoint subsets: the first subset is used to build the SMs, whereas the second is used to evaluate the precision of the generated SMs. This process is repeated multiple times as the training dataset is randomly partitioned. In order to assess the model precision, the absolute value of the differences between the observed $o(x_i)$ and the predicted $p(x_i)$ application output from the SM is computed by , where n' is the number of points for testing in the cross-validation. The SM that performs the best is then built using the entire training dataset.

2.4. Accelerating the surrogate model building phase with parallel execution

SMs can significantly reduce the number of workflow runs in SA, but a target application still has to be executed to generate a training dataset. This may take a long time. In order to accelerate application runs, we use a runtime system called Region Templates (RT) [18], which enables the execution of image analysis applications on hybrid distributed memory machines equipped with accelerators (e.g., GPUs and Intel Phi co-processors). An application is described in the RT framework as a multi-level dataflow in which each coarse-grain application stage may be represented as another fine-grain dataflow. This multi-level representation enables RT to apply different task assignment strategies at each level, improving flexibility on hybrid machines.

The stages of an application workflow communicate through data elements called data regions in RT. Instead of sending information directly from/to computation stages, a stage produces and consumes data regions, whereas the transfers of the data regions among the computing devices on the machine are handled by the RT runtime system. This strategy enables the transparent placements of the application on parallel machines to reduce data transfer demands. More details on the scheduling strategies and data-aware placement in the RT framework are available in our previous work [18].

We have also developed an optimization in RT for SA studies, called Reuse Tree Merging (RTM) strategy, that enables reuse of repeated



```
SIZE,ELONG,ROUND,FLATN..
26, 1.75, 0.74, 1.44
44, 1.81, 0.65, 1.33
36, 1.13, 0.77, 1.27
23, 1.24, 0.51, 1.27
57, 1.61, 0.58, 1.48
41, 1.69, 0.78, 1.50
47, 1.66, 0.56, 1.57
         .
         .
```

Fig. 2. Morphological correlation framework. Morphological features computed from WSIs and aggregated into a patient basis to create patient signatures. The signatures are then used to group patients, and the significance of the survival in these groups are assessed.

computations from multiple application runs [39]. As input parameters are varied in a SA, there may be parts of the application workflow that use the same parameter values and input data, which can be reused and prevented from executing multiple times. These reuse opportunities appear on coarse-grain stages of the application and on fine-grain tasks within a stages. Our approaches are able to take advantage and improve performance on both cases.

3. Results

Our experiments used whole slide tissue image datasets from Breast Invasive Carcinoma (BRCA), Glioblastoma Multiforme (GBM), Low Grade Glioma (LGG), and Lung Squamous Cell Carcinoma (LUSC) cases from The Cancer Genome Atlas repository.[1] We partitioned the images into 4 K × 4 K tiles for parallel computation. We executed the large scale experiments on a distributed memory parallel machine. Each node of this machine has two Xeon E5-2680 8-core Sandy Bridge processors and one Intel Xeon Phi SE10P coprocessor. We carried out small scale experiments after the generation of the training dataset on a local machine with an Intel (R) Core (TM) E5-2640 CPU with 2.60 GHz, 64 GB of RAM. All of the machines run Linux OS. We evaluated the following SMs: Gaussian Process (GP), MARS Cubic (MARS-C), MARS Linear (MARS-L), Artificial Neural Network (NN), Polynomial Linear (PL), Polynomial Quadratic (PQ), and Radial Basis Function (RBF).

This section is organized as follows. First, we present the correlation analysis application studied in this work in Section 3.1. We then evaluate the accuracy of the SMs in SA in Sections 3.2–3.5. These experiments employ a single WSI per tissue type because a large number of executions are performed. In Sections 3.6 and 3.7, we execute SA studies using 55 WSIs to demonstrate the scalability of our solution and a comparison to a previous work that executed SA without the use of SMs. Finally, in Section 3.8, we carry out a correlation analysis with 381 patients/WSIs with Lung Squamous Cell Carcinoma (LUSC) from the TCGA dataset. This experiment analyzes the impact of input parameters in segmentation algorithms in downstream analyses.

3.1. Morphological correlation

This section briefly describes morphological correlation analysis used in this work. In the correlation analysis, nuclei are segmented in WSIs, quantitative features are computed per nucleus (set of 60 features from 20 quantile normalized features as listed in Table A.2), and the features are aggregated per patient to create a morphological patient signature. The signatures are then clustered with a consensus clustering [40] for a robust grouping. This type of correlation analysis has been executed by multiple studies [16,41].

This work evaluates parameters of the segmentation phase of our application, which is the most parameterized step of workflow. However, after identifying important parameters in the segmentation, they are varied with the goal of evaluating the propagation of uncertainty in the segmentation results to the overall correlation analysis (i.e., patient clustering and survival analysis). The complete framework is presented in Fig. 2. The details of each analysis phase are provided in Section S1 in the supplementary materials.

3.2. Performance of surrogate models in a SA study

This section evaluates the SMs in a SA study with the watershed workflow and GBM images. The SMs were created using a training dataset of 100 elements, and were used in a SA to compute Pearson's Correlation Coefficients in which the SMs are executed for 1000 parameter sets. For the sake of comparison, we performed the same SA study using the actual application (AA).

The results are presented in Fig. 3, which shows the impact of each parameter on the output. The SMs that attained indices most similar to AA were MARS-L, MARS-C, and GP. In order to assess the performance of each SM, we computed the Mean Absolute Error (MAE) [42,43] of the parameters coefficients obtained with the SM and the actual application. These results are presented in Fig. 4 for training datasets with 100, 200, and 300 elements. The Polynomial Quadratic (PQ) results are not shown for 100 points because it requires a minimum training set with 200 elements.

The results confirm that MARS-L, MARS-C, and GP as the best performing SMs. The high MAE values for NN, PL, and RBF clearly show that these SMs are not good candidates in our case. RBF is interesting since it estimates coefficient values close to zero for all parameters. Thus, although the RBF's MAE values are smaller than those of NN and PL, it has not generated useful coefficients. To investigate if NN, PL, and RBF have attained low performance because of the training dataset, we increased the training size to 10,000 elements. We then noticed that those SMs improved their performance with larger training datasets, but they were still not superior than other models (e.g. MARS-L, MARS-C, and GP). We should note that if large training datasets are used, the benefits from SMs may be offset by the cost of creating the training datasets.

3.3. Evaluating SA and SMs on multiple cancer types

This section extends our evaluation with the use of multiple cancer types (GBM, BRCA, LGG, and LUSC) and the watershed segmentation workflow. These experiments used the three best models identified in the previous results: GP, MARS-L, and MARS-C, and a training dataset of 100 elements.

The results in Fig. 5 show that the SMs approximate the AA well for all cancer types, regardless of the significant variation of the correlation values in intensity and direction across tissue types. We computed the MAE for all of the SMs and cancer types. The MAEs are smaller than 0.04 for the selected SMs. Our experiments also show that changing a parameter value may have a different impact on segmentation results for different tissue/cancer types. We should note that the differences in parameter values are not meant to distinguish between cancer types but

[1] https://cancergenome.nih.gov/.

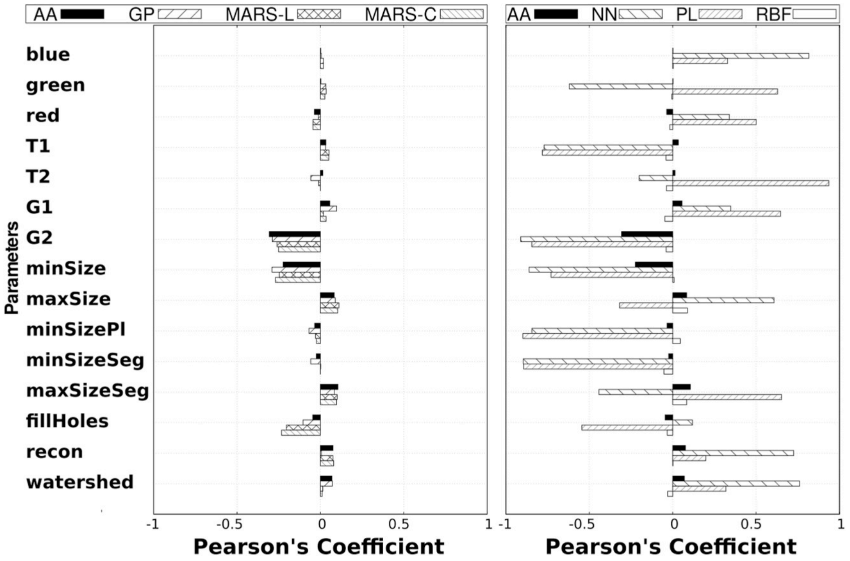


Fig. 3. Pearson's Coefficient using SMs and the actual application for GBM images.

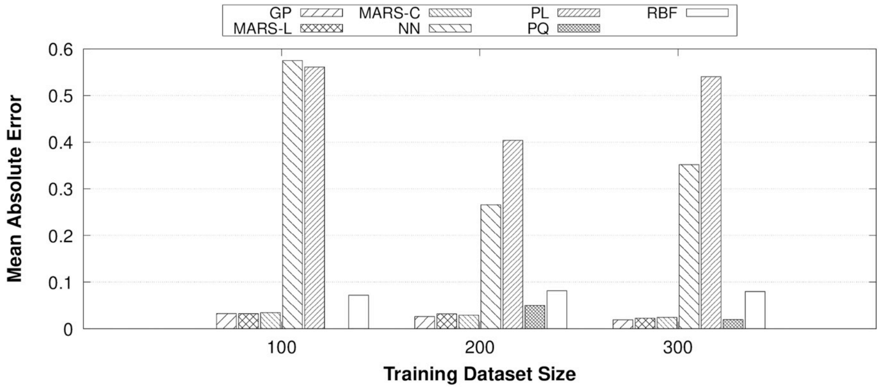


Fig. 4. Mean Absolute Error among Pearson's Coefficient generated using SMs and the actual application for GBM images.

to measure the difference in impact of the parameter values on the segmentation results across cancer types.

3.4. Performance of SMs on the VBD SA

This section evaluates the SMs on VBD SA studies with the watershed workflow. We used GP, MARS-L and MARS-C and training datasets with 100–300 points. The SMs were employed to execute a VBD study that uses $n = 300$ for a total of $n(k + 2)$ application runs. Given that $k = 15$ parameters exist, it requires 5,100 runs. Because of the high computation demands to execute the study using the actual application (shown as VBD) for comparison purpose, we used only the GBM images.

Fig. 6 shows main and total effects of each parameter with the SMs as compared to the VBD using the application. The SMs were built with a training dataset of 100 elements. The models were able to approximate well the effects and only small differences among the actual VBD and the SMs based SA were observed. Further, we also increased the training size used to build the SMs and measured the MAE for each SM. As is shown in Fig. 7, the larger training sizes tend to improve the precision of the study, but the observed errors are small even when 100 samples are used. The MAE values are below 0.022 for GBM. When using a training dataset size of 100, we were able to speed up the VBD study by 51 × compared to the VBD using the actual application.

3.5. SA using level set based segmentation workflow

This experiment evaluates the SMs in VBD studies using the level set based segmentation workflow. We have employed GBM to study the $k = 6$ parameters of the application (Table A1). Although this workflow has a smaller number of parameters as compared to the watershed based one, the SA study in this case is more costly because of the higher workflow execution time. The VBD used $n = 300$ that results in $n(k + 2) = 2{,}400$ application (or SM) runs. The SMs were built using training datasets with 100 elements.

The results of the VBD computed using the actual application (shown as VBD), and those employing the SMs are presented in Fig. 8 for each parameter. As shown, the SMs slightly overestimated the impact of the most important parameter (OTSU) with respect to its main effect, whereas the total effects that include higher-order interactions were better approximated. In all cases, the order or ranking of parameters importance was maintained for significant parameters. The SMs analysis executed about 24 × faster compared to VBD using the actual application.

Fig. 9 presents the MAE for SMs as the training dataset size is varied. The error is small for all models. The increasing in the dataset size has little impact on the results. We attribute the small improvement in this case to the fact that a large fraction of the output variation in the segmentation workflow is determined by one or two parameters (OTSU and CW). Hence, the variation could be more easily approximated as compared to the watershed based workflow.

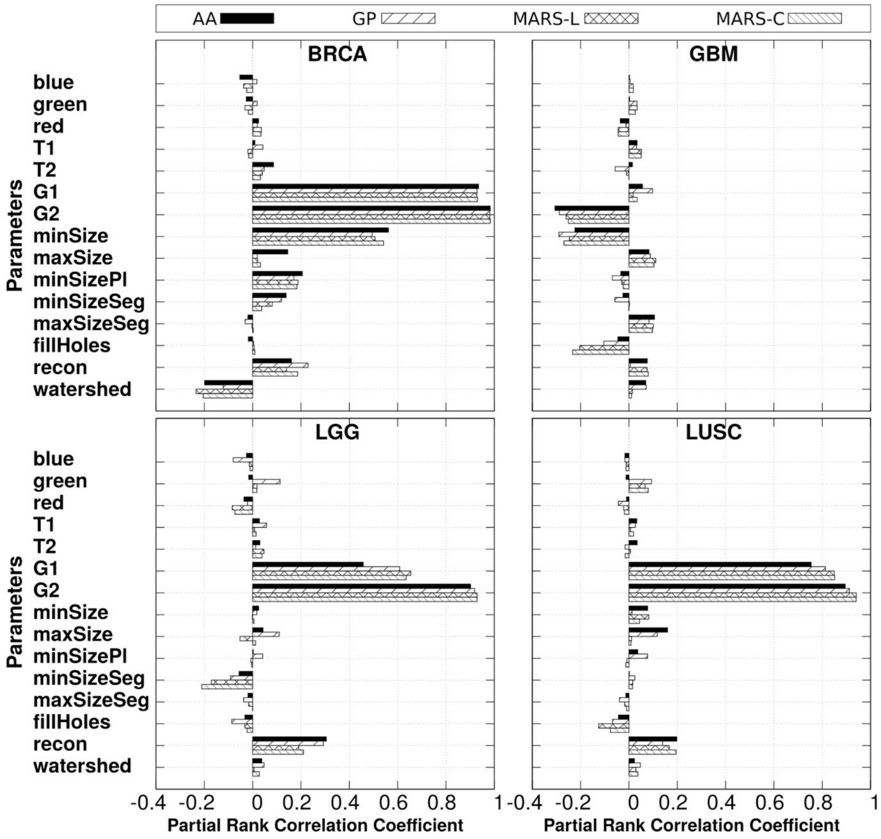


Fig. 5. SM efficiency evaluation for different types of cancer.

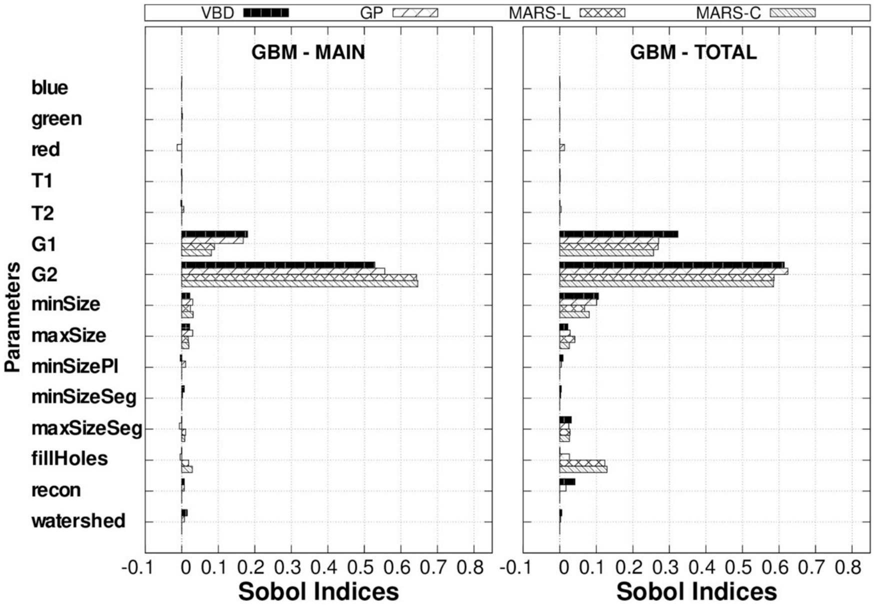


Fig. 6. SMs efficiency for the VBD analysis.

3.6. Scalability evaluation

This section presents the execution on distributed hybrid machines equipped with CPUs and Intel Phi. The watershed workflow and dataset with 55 WSIs or 6113 4 K × 4 K GBM image tiles were used. The application workflow is composed of normalization, segmentation, and comparison stages. Fig. 10 presents the execution time per parameter set evaluated as the number of nodes used is increased. The parallel efficiency of the execution is over 0.92. When the CPU and Intel Phi are used in coordination, there is a performance gain of about 2.28 × as compared to the CPU-only using 16 CPU-cores available in each node. In addition, we evaluate the computation reuse strategies: Coarse-Grain Reuse (CGR), proposed in our early work [44], that only reuses a stage if all parameters of the stage have the same values and the Reuse-Tree Merging (RTM) that is able to reuse

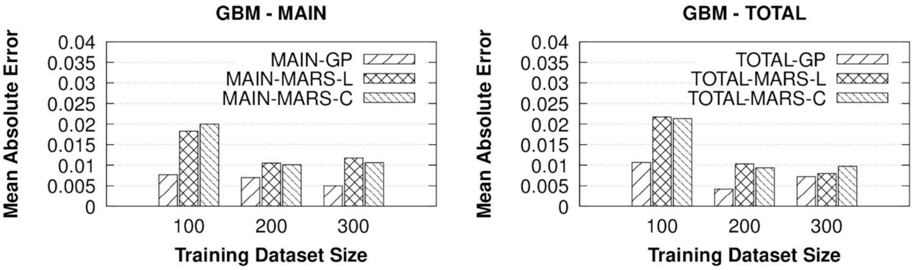


Fig. 7. Mean absolute error for different types of cancer in VBD analysis.

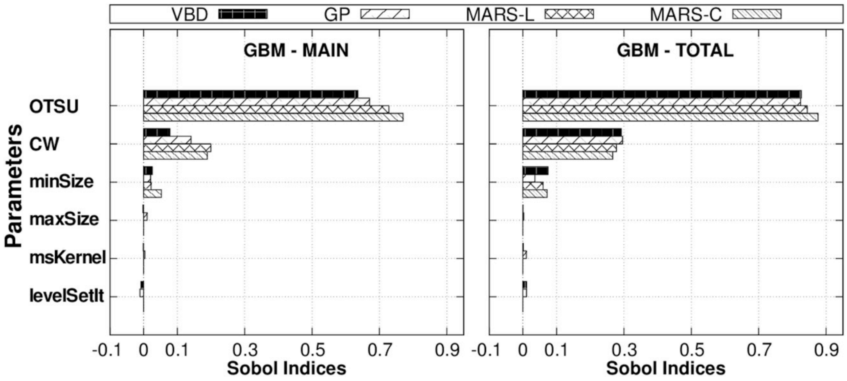


Fig. 8. SMs efficiency for VBD and the level set based workflow.

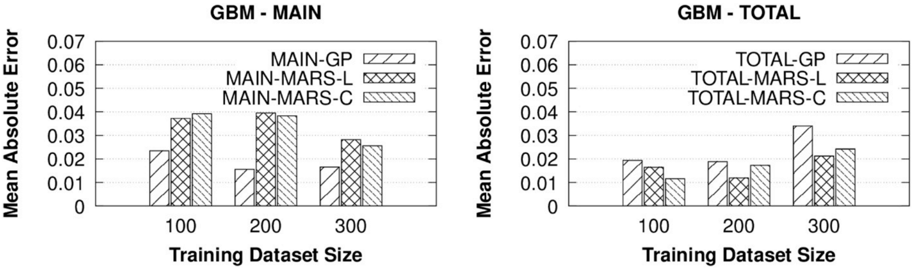


Fig. 9. MAE of the level set based workflow for VBD.

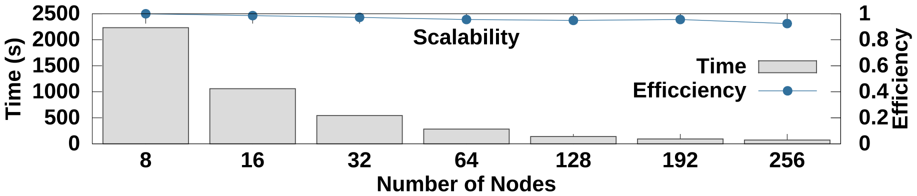


Fig. 10. Distributed workflow execution on hybrid nodes equipped with CPU and Intel Phi.

computation in fine-grain tasks within a stage even if only a subset of the parameters are the same. As is shown in Table 1, the CGR optimization significantly improved the execution without computation reused (CPU + Phi), whereas the RTM further accelerated the SA execution in about 1.5 ×
.

Table 1
Execution time with 256-nodes and different optimizations.

CPU-only(s)	CPU + Phi(s)	CPU + Phi + CGR(s)	CPU + Phi + CGR(s)
72.69	31.84	19.90	13.22

3.7. *Large-scale SA with vs without SMs*

This experiment compares the SM based SA strategy to a large-scale SA that does not employ SMs, which was carried out on our previous work [44]. The experiment executed a VBD study with the watershed workflow and 55 WSIs of GBM. The study has $k = 8$ parameters only, since the other ones were identified as non-influential and previously pruned with MOAT. As a result, $N = n(k + 2)$ or 2000 workflow runs are required for the $n = 200$ used. Higher values of n were not used due to the cost of the analysis – the experiment took 42.9 h on 128 nodes and involved significant I/O demands as 820 Terabytes of data were

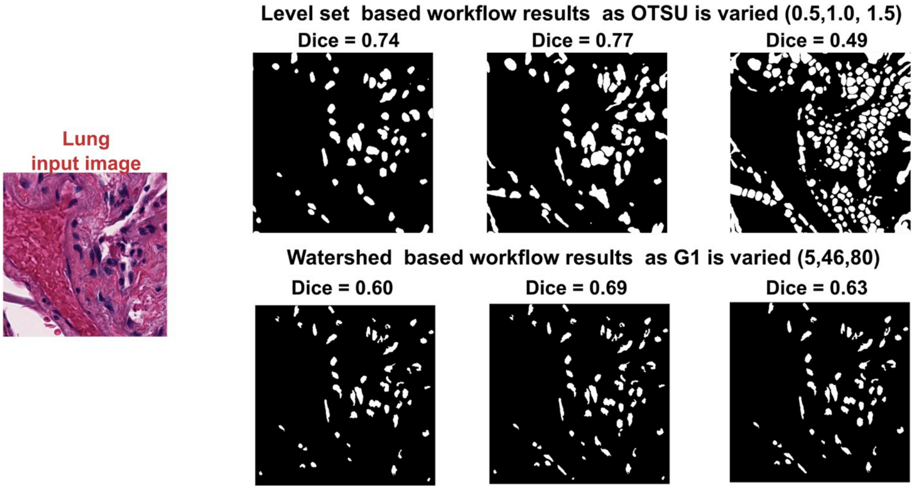
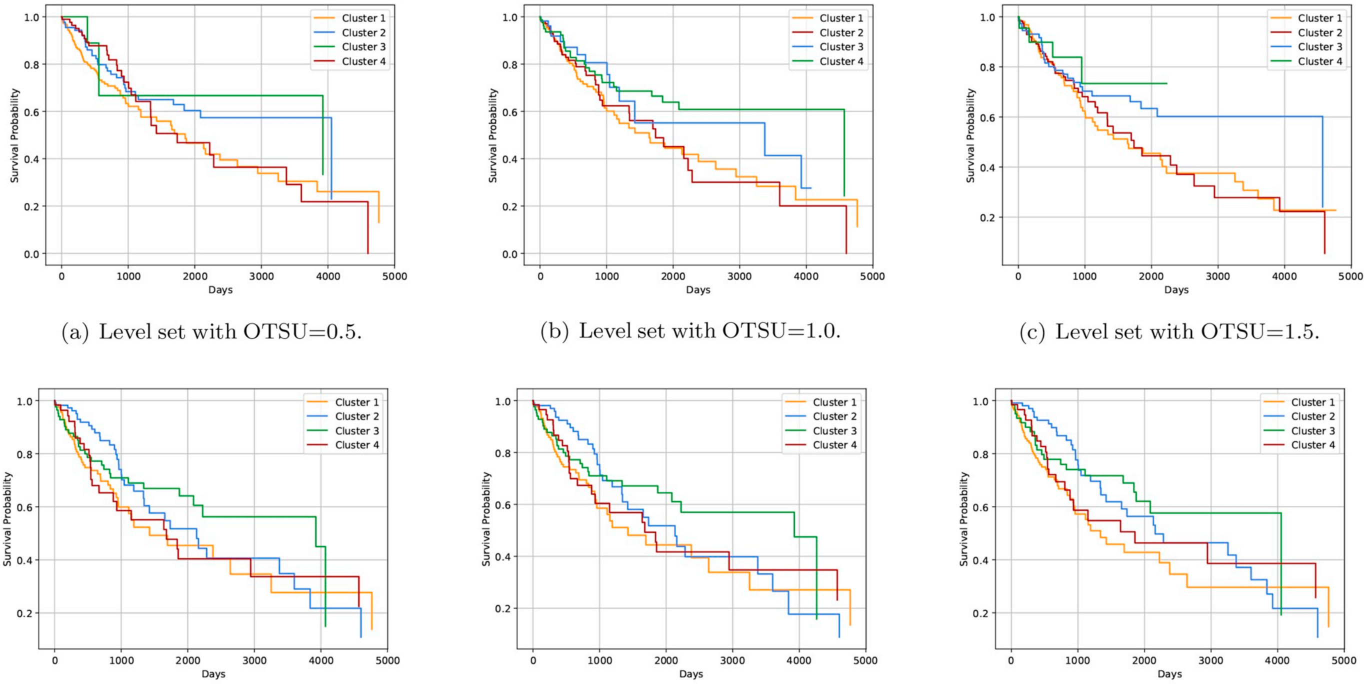

Level set based workflow results as OTSU is varied (0.5,1.0, 1.5)
Dice = 0.74
Dice = 0.77
Dice = 0.49
Lung
input image
Watershed based workflow results as G1 is varied (5,46,80)
Dice = 0.60
Dice = 0.69
Dice = 0.63

Fig. 11. Segmentation for both workflows and the parameter values used in the survival analysis with an example image patch. Ground-truth segmentation for Dice calculation was generated by a pathologist.



(a) Level set with OTSU=0.5.

(b) Level set with OTSU=1.0.

(c) Level set with OTSU=1.5.

(d) Watershed with G1=5.

(e) Watershed with G1=46.

(f) Watershed with G1=80.

Fig. 12. Kaplan-Meier survival estimations using patient morphological signatures computed with different segmentation algorithms and parameter values.

produced and processed.

We executed the same analysis using SMs in which a training dataset size of 100 was used to build a GP SM. This reduced the number of runs by 20 × as compared to the original experiment [44]. The execution time dropped to 2.14 h using the same number of nodes, and the I/O was reduced to about 40 Terabytes. These performance gains were achieved with a MAE of only 0.02 and 0.04, respectively, for the main and total effects.

3.8. Impact of segmentation to correlative survival analysis

This section evaluates the impact of the variations in the segmentation results due changes in important parameters to a correlative survival analysis. We varied the significant parameters of the watershed (G1) and level set (OTSU) based segmentation workflows identified in the SA. We the parameter values around manually tuned values selected by the developers. As such, the central point of our analysis using OTSU = 1.0 and G1 = 46 are the manually selected values, which were increased/decreased while keeping all the other parameters fixed. We first present in Fig. 11 the segmentation results for each parameter configuration. The algorithms were differently affected by the parameter value changes. The Level set workflow attained higher Dice values than the Watershed workflow. The Watershed workflow had a smaller variation in the segmentation results.

The consensus clustering was applied to patient level morphological signatures compute for each parameter chosen. Note that this analysis executes the actual segmentation workflows with the TCGA images without using the surrogates. The SMs were not used in these experiments because as modeled in this paper they are not capable of generating segmentation results (masks) used in the correlative analysis.

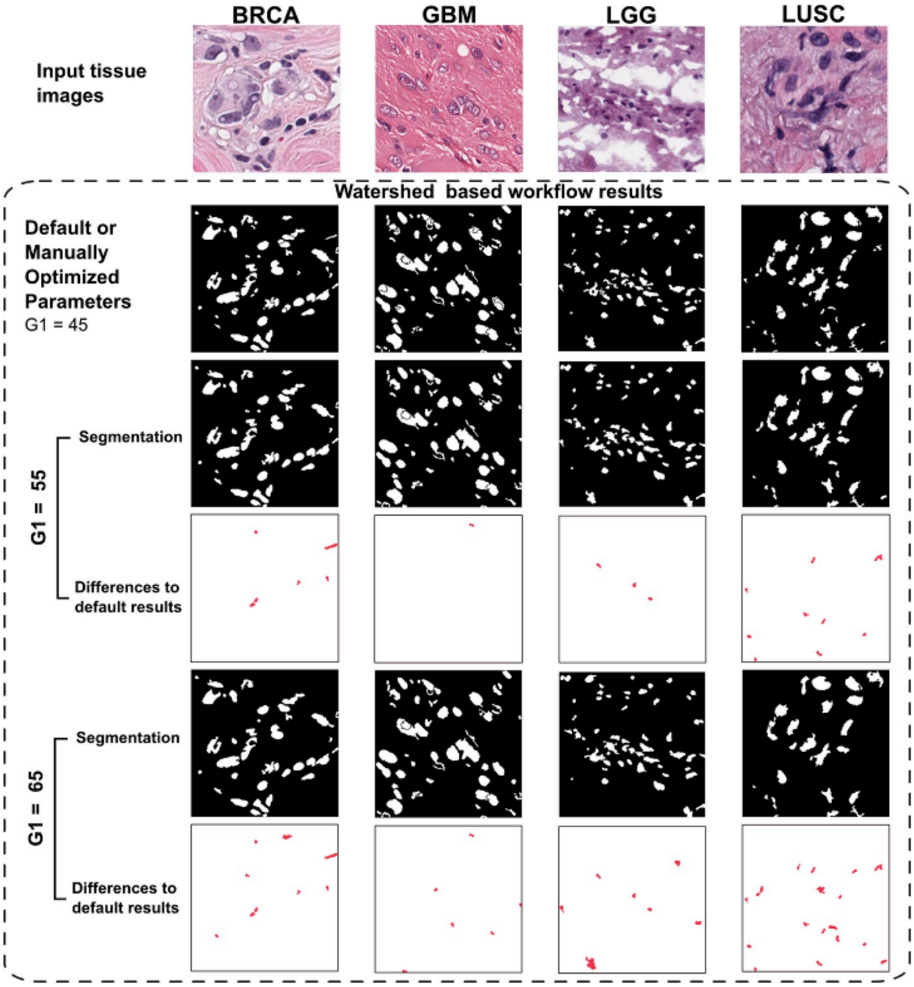

BRCA
GBM
LGG
LUSC
Input tissue images
Watershed based workflow results
Default or Manually Optimized Parameters
G1 = 45
G1 = 55
Segmentation
Differences to default results
G1 = 65
Segmentation
Differences to default results

Fig. 13. Image patches (300 × 300 pixels) from the used cancer types. Segmentation results as the *G*1 parameter is changed shown along with the differences in the segmentation as compared to the manually optimized parameters for each parameter value.

Instead, in this work, the SMs were employed to measure only how DICE values vary as parameters change. The number of clusters was varied from two to seven. We present the results for four clusters, which attained the most significant separation in the survival according to the logrank test. This analysis uses 381 patients/WSIs with Lung Squamous Cell Carcinoma (LUSC) from the TCGA dataset. The patient information is provided in Table A.3.

The Kaplan-Meier survival estimations for both segmentation algorithms and multiple parameter values are shown in Fig. 12. The parameter variations lead to significantly different survival curve shapes for both algorithms. The Fowlkes-Mallows index [45] was used to assess the agreement among clusterings for each algorithm using different parameter values. For the level set based segmentation the index was only 0.43, whereas it was 0.72 for the watershed segmentation. This highlights important changes in patient grouping.

The logrank test was used to evaluate if the clusters have significantly different outcomes, and how parameters changes impact them. In the Level set case, only the Clusters 1 and 3 using OTSU = 1.0 (Fig. 12(a)) are significantly different (P = 0.02), whereas other parameter values have generated indistinguishable clusters (smallest P-values found among all pairs of clusters were 0.16 for OTSU = 0.5 and 0.46 for OTSU = 1.5). In the watershed case, the survival analysis is more stable. For instance, regardless of the parameter value used, it is possible to distinguish at least between Clusters 1 and 2. We attribute this to the smaller variations in the segmentation output from this algorithm, as the input parameters are varied. However, the parameter value used still impacts the significance between clusterings survival. For instance, Clusters 2 and 4 have a more significant separation with G1 = 5 (P = 0.06) as compared to G1 = 80 (P = 0.18).

4. Discussion

In an increasing number of research studies Pathology imaging features have been used to investigate relationships and correlations between disease morphology and genomics signatures and clinical outcome [7,16,46–52]. While all these works have made significant discoveries using imaging features in basic, translational and clinical research, neither of them have performed SA to investigate the stability or uncertainty of the results/discoveries. However, we have shown that changes in the segmentation parameters of correlative analysis workflows may lead to significant variations in conclusions, for instance, as presented in our survival analyses.

The use of SMs has been proposed here as a strategy to reduce the computation challenges of carrying out SA in the pathology image analysis domain. As compared to the previous work [44] that compute SA without the use of SMs, its use has reduced the computational cost in up to 51 × with a small penalty to the SA indices. SMs have been used before in other domains [9–12,53,53]. Here, we adapt and evaluate these strategies in a novel domain and show the benefits of their use.

Most of the related works have employed a single SM for a specific application. In Ref. [54], however, multiple SM strategies are combined to perform a better fitting of a nuclear system modeling application, so that each SM strategy is used for a given parameter space subregion. The experiments do not compare the SMs individually, but they show that GP and MARS are selected for most of the subregions since they have the overall best performance. These results agreed with our observations as GP and MARS have also attained the best results in our domain. Other works have had multiple challenges in achieving using SMs based on NN [55] and RBF [56] due to aspects that include complexity of tuning the strategy and domain dimensionality.

We also show that impact of the parameters in the segmentation results vary according to the cancer type evaluated (Section 3.3). To show the impact of parameters across image types, we extracted a small patch from each image type (Fig. 13) in our dataset. The $G1$ parameter from the watershed workflow, for instance, has a smaller impact in GBM. It is used for candidate object set identification. Because there is high color difference between the boundary and interior of a nucleus in GBM images, the thresholds performed with G1 with respect to such variations tend to identify foreground pixels similarly regardless of the parameter values used. To show this in practice, we present segmentation results in Fig. 13 as the $G1$ value is changed. For each G1 value, the segmentation results computed and the differences from these results to the ones using the default (manually optimized) parameters are shown. The changes in the GBM results (shown in red in the difference image) are smaller than those observed in other cancer types, agreeing with the SA indices.

For both workflows, parameters associated with the candidate object detection phase have been identified as the most important in the SA. Previous works [57–59] have highlighted that the robustness of this phase of the application is essential to attain good segmentation results. Here, on the other hand, we quantify the effect to the overall segmentation results due to parameter changes on that phase (and others). We noticed that the problem with this phase is not only in the differences in the number of objects found (e.g., 18% smaller for G2 = 80 as compared to G2 = 5 with the watershed workflow), but also with the initial shape of the detected object that will significantly affect the final shape of the objects segmented and features computed for them.

An important limitation of our work is that while we are able to measure parameter sensitivity in the segmentation phase and we have shown that it significantly impacts other phases of the application (correlative analysis), our current studies and methods can not systematically measure the propagation path from the segmentation to the correlative results. We imagine that being able to identify the dynamics and effects of the uncertainty propagation over all phases of the application is important to mitigate its impact. This is one of our main future research directions.

5. Conclusions

We have evaluated the use of SMs to accelerate SA studies in microscopy image segmentation workflows. In the analysis using the watershed based workflow, the SMs based VBD SA improved the study performance in about 51 × and computed SA indices with an error (MAE) of only about 0.022. For the level set based segmentation workflow, our approach achieved performance improvements of about 24 × , while the MAE remains about 0.02. We have also shown that changes in important parameters of the segmentation, identified in the SA studies, significantly affect other phases of application, such as, downstream correlation analysis. Furthermore, with the increasing use of pathology image analysis, the techniques and strategies proposed have potential to enable more routinely use of SA in the domain.

Conflicts of interest

There are no conflicts of interest in this study from any of the authors.

Acknowledgment

This work was supported in part by U24CA180924, U24CA215109, 1UG3CA225021 from the NCI, R01LM011119-01 and R01LM009239 from the NLM, CNPq, and NIH K25CA181503. This research used resources of the XSEDE Science Gateways program under grant TG-ASC130023.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compbiomed.2019.03.006.

References

[1] O. Steichen, C.D.L. Bozec, M. Thieu, E. Zapletal, M.-C. Jaulent, Computation of semantic similarity within an ontology of breast pathology to assist inter-observer consensus, Comput. Biol. Med. 36 (7) (2006) 768–788 (special Issue on Medical Ontologies).

[2] M. Gadermayr, D. Eschweiler, A. Jeevanesan, B.M. Klinkhammer, P. Boor, D. Merhof, Segmenting renal whole slide images virtually without training data, Comput. Biol. Med. 90 (2017) 88–97.

[3] A. Mezheyeuski, I. Hrynchyk, M. Karlberg, A. Portyanko, L. Egevad, P. Ragnhammar, D. Edler, B. Glimelius, A. Östman, Image analysis-derived metrics of histomorphological complexity predicts prognosis and treatment response in stage II-III colon cancer, Sci. Rep. 6 (2016) 36149 (EP –).

[4] S. Kothari, J.H. Phan, T.H. Stokes, M.D. Wang, Pathology imaging informatics for quantitative analysis of whole-slide images, J. Am. Med. Inform. Assoc. : JAMIA 20 (6) (2013) 1099–1108.

[5] Lee Cooper, et al., Integrated morphologic analysis for the identification and characterization of disease subtypes, J. Am. Med. Inform. Assoc. 19 (2) (2012) 317–323.

[6] J. Kong, L.A.D. Cooper, F. Wang, J. Gao, G. Teodoro, L. Scarpace, T. Mikkelsen, M.J. Schniederjan, C.S. Moreno, J.H. Saltz, D.J. Brat, Machine-based morphologic analysis of glioblastoma using whole-slide pathology images uncovers clinically relevant molecular correlates, PLoS One 8 (11) (2013) 1–17.

[7] M.N. Gurcan, L. Boucheron, A. Can, A. Madabhushi, N. Rajpoot, B. Yener, Histopathological image analysis: a review, IEEE reviews in biomedical engineering 2 (2009) 147–171.

[8] Y. Gao, V. Ratner, L. Zhu, T. Diprima, T. Kurc, A. Tannenbaum, J. Saltz, Hierarchical nucleus segmentation in digital pathology images, Proc. SPIE 9791 (2016) 979117–979117–6.

[9] Queipo, V. Nestor, T. Haftka Raphael, Wei Shyy, Tushar Goel, Rajkumar Vaidyanathan, P Kevin Tucker, Surrogate-based analysis and optimization, Prog. Aero. Sci. 41 (1) (2005) 1–28.

[10] W. Shyy, N. Papila, R. Vaidyanathan, K. Tucker, Global design optimization for aerodynamics and rocket propulsion components, Prog. Aero. Sci. 37 (1) (2001) 59–118.

[11] X. Wan, J.F. Pekny, G.V. Reklaitis, Simulation-based optimization with surrogate modelsApplication to supply chain management, Comput. Chem. Eng. 29 (6) (2005) 1317–1328.

[12] D.W. Stephens, Dirk Gorissen, Karel Crombecq, Tom Dhaene, Surrogate based sensitivity analysis of process equipment, Appl. Math. Model. 35 (4) (2011) 1676–1687.

[13] S. Ramtani, Parametric sensitivity analysis applied to a specific one-dimensional internal bone remodelling problem, Comput. Biol. Med. 37 (8) (2007) 1203–1209.

[14] T. Wang, F. Liang, Z. Zhou, X. Qi, Global sensitivity analysis of hepatic venous pressure gradient (HVPG) measurement with a stochastic computational model of the hepatic circulation, Comput. Biol. Med. 97 (2018) 124–136.

[15] B.M. Johnston, P.R. Johnston, Sensitivity analysis of ST-segment epicardial potentials arising from changes in ischaemic region conductivities in early and late stage ischaemia, Comput. Biol. Med. (2018) 288–299.

[16] R.G. Verhaak, et al., An integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR and NF1, Cancer Cell 17 (1) (2010) 98.

[17] L.A. Cooper, J. K, et al., Integrated morphologic analysis for the identification and characterization of disease subtypes, J. Am. Med. Inform. Assoc. 19 (2) (2012) 317–323.

[18] G. Teodoro, T. Pan, T. Kurc, J. Kong, L. Cooper, S. Klasky, J. Saltz, Region templates: data representation and management for high-throughput image analysis, Parallel Comput. 40 (10) (2014) 589–610.

[19] L.R. Dice, Measures of the amount of ecologic association between species, Ecology 26 (3) (1945) 297–302.

[20] P. Jaccard, Etude comparative de la distribution florale dans une portion des Alpes et du Jura, Impr. Corbaz, 1901.

[21] M.D. Morris, Factorial sampling plans for preliminary computational experiments, Technometrics 33 (2) (1991) 161–174.

[22] A. Saltelli, S. Tarantola, F. Campolongo, M. Ratto, Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models, Wiley, 2004.

[23] V.G. Weirs, J.R. Kamm, L.P. Swiler, S. Tarantola, M. Ratto, B.M. Adams, W.J. Rider, M.S. Eldred, Sensitivity analysis techniques applied to a system of hyperbolic conservation laws, Reliab. Eng. Syst. Saf. 107 (2012) 157–170.

[24] J. Morio, Global and local sensitivity analysis methods for a physical system, Eur. J. Phys. 32 (6) (2011) 1577–1583.

[25] I.M. Sobol, Sensitivity estimates for nonlinear mathematical models,

Matematicheskoe Modelirovanie 2 (1990) 8.

[26] A. Saltelli, Making best use of model evaluations to compute sensitivity indices, Comput. Phys. Commun. 145 (2) (2002) 280–297.

[27] B. Iooss, P. Lemaitre, A review on global sensitivity analysis methods, in: G. Dellino, C. Meloni (Eds.), Uncertainty Management in Simulation-Optimization of Complex Systems, Vol. 59 of Operations Research/Computer Science Interfaces Series, Springer, US, 2015, pp. 101–122.

[28] I. Sobol, Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates, Math. Comput. Simulat. 55 (1–3) (2001) 271–280.

[29] T. Homma, A. Saltelli, Importance measures in global sensitivity analysis of nonlinear models, Reliab. Eng. Syst. Saf. 52 (1) (1996) 1–17.

[30] Bruno Sudret, Stefano Marelli, Joe Wiart, Surrogate models for uncertainty quantification: an overview, Antennas and Propagation (EUCAP), 2017 11th European Conference on, IEEE, 2017, pp. 793–797.

[31] Bartz-Beielstein T Naujoks B Stork J Zaefferer M, Tutorial on surrogate-assisted modelling, Tech. Rep. D.12, Synergy for Smart Multi-Objective Optimisation (October 2016).

[32] Nicholas J. Higham, Accuracy and Stability of Numerical Algorithms, SIAM, 2002.

[33] Dishi Liu, Hermann G. Matthies, Pivoted Cholesky Decomposition by Cross Approximation for Efficient Solution of Kernel Systems, (2015), p. 19 1505.06195.

[34] H.-M. Gutmann, A radial basis function method for global optimization, J. Glob. Optim. 19 (2001) 201–227.

[35] Yew S. Ong, Prasanth B. Nair, Andrew J. Keane, Evolutionary optimization of computationally expensive problems via surrogate modeling, AIAA J. 41 (4) (2003) 687–696.

[36] Jerome H. Friedman, Multivariate adaptive regression splines, Ann. Stat. (1991) 1–67.

[37] B. Yegnanarayana, Artificial Neural Networks, PHI Learning Pvt. Ltd., 2009.

[38] Steve R. Gunn, Support vector machines for classification and regression, others, ISIS technical report 14 (1998) 85–86.

[39] W. Barreiros, G. Teodoro, T. Kurc, J. Kong, A.C.M.A. Melo, J. Saltz, Parallel and efficient sensitivity analysis of microscopy image segmentation workflows in hybrid systems, 2017 IEEE International Conference on Cluster Computing (CLUSTER), 2017, pp. 25–35.

[40] Stefano Monti, Pablo Tamayo, Jill Mesirov, Todd Golub, Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data, Mach. Learn. 52 (1–2) (2003) 91–118.

[41] Lee AD. Cooper, Jun Kong, Fusheng Wang, Tahsin Kurc, Moreno, S. Carlos, Daniel J. Brat, Joel H. Saltz, Morphological signatures and genomic correlates in glioblastoma, 2011 IEEE International Symposium on Biomedical Imaging, From Nano to Macro, 2011, pp. 1624–1627.

[42] Tianfeng Chai, Roland R. Draxler, Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature, Geosci. Model Dev. (GMD) 7 (3) (2014) 1247–1250.

[43] Cort J. Willmott, Kenji Matsuura, Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance, Clim. Res. 30 (1) (2005) 79–82.

[44] G. Teodoro, T.M. Kur, L.F.R. Taveira, A.C.M.A. Melo, Y. Gao, J. Kong, J.H. Saltz, Algorithm sensitivity analysis and parameter tuning for tissue image segmentation pipelines, Bioinformatics 33 (7) (2017) 1064–1072.

[45] E.B. Fowlkes, C.L. Mallows, A method for comparing two hierarchical clusterings, J. Am. Stat. Assoc. 78 (383) (1983) 553–569.

[46] L.A. Cooper, D.A. Gutman, C. Chisolm, C. Appin, J. Kong, Y. Rong, T. Kurc, E.G.V. Meir, J.H. Saltz, C.S. Moreno, D.J. Brat, The tumor microenvironment strongly impacts master transcriptional regulators and gene expression class of glioblastoma, Am. J. Pathol. 180 (5) (2012) 2108–2119.

[47] P. Mobadersany, S. Yousefi, M. Amgad, D.A. Gutman, J.S. Barnholtz-Sloan, J.E. Velázquez Vega, D.J. Brat, L.A.D. Cooper, Predicting cancer outcomes from histology and genomics using convolutional networks, Proc. Natl. Acad. Sci. Unit. States Am. 115 (13) (2018) E2970–E2979.

[48] A.H. Beck, A.R. Sangoi, S. Leung, R.J. Marinelli, T.O. Nielsen, M.J. van de Vijver, R.B. West, M. van de Rijn, D. Koller, Systematic analysis of breast cancer morphology uncovers stromal features associated with survival, Sci. Transl. Med. 3 (108) (2011) 108ra113–108ra113.

[49] T.J. Fuchs, P.J. Wild, H. Moch, J.M. Buhmann, Computational pathology analysis of tissue microarrays predicts survival of renal clear cell carcinoma patients, in: D. Metaxas, L. Axel, G. Fichtinger, G. Székely (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2008, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 1–8.

[50] K.-H. Yu, C. Zhang, G.J. Berry, R.B. Altman, C. Ré, D.L. Rubin, M. Snyder, Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features, Nat. Commun. 7 (2016) 12474 12474–12474.

[51] M. Veta, R. Kornegoor, A. Huisman, A.H.J. Verschuur-Maes, M.A. Viergever, J.P.W. Pluim, P.J. van Diest, Prognostic value of automatically extracted nuclear morphometric features in whole slide images of male breast cancer, Mod. Pathol. 25 (2012) 1559 (EP –).

[52] J.L. Carstens, P. Correa de Sampaio, D. Yang, S. Barua, H. Wang, A. Rao, J.P. Allison, V.S. LeBleu, R. Kalluri, Spatial computation of intratumoral T cells correlates with survival of patients with pancreatic cancer, Nat. Commun. 8 (2017) 13.

[53] N.S. Ribeiro, J. Folgado, H.C. Rodrigues, Surrogate-based visualization and sensitivity analysis of coronary stent performance: a study on the influence of geometric design, International Journal for Numerical Methods in Biomedical Engineering 34 (10) (2018) e3125.

[54] G.A. Banyay, S.D. Smith, J.S. Young, Sensitivity Analysis of a Nuclear Reactor System Finite Element Model vol. 10, (2018) (40795).

[55] A. Kaveh, S.M. Hamze-Ziabari, T. a. Bakhshpoori, ESTIMATING DRYING SHRINKAGE OF CONCRETE USING A MULTIVARIATE ADAPTIVE REGRESSION SPLINES APPROACH, International Journal of Optimization in Civil Engineering 8 (2) (2018) 181–194.

[56] Z. Wu, W. Wang, D. Wang, K. Zhao, W. Zhang, Global sensitivity analysis using orthogonal augmented radial basis function, Reliab. Eng. Syst. Saf. 185 (2019) 291–302.

[57] C. Couprie, L. Najman, H. Talbot, Seeded segmentation methods for medical image analysis, in: G. Dougherty (Ed.), Medical Image Processing, Biological and Medical Physics, Biomedical Engineering, Springer, New York, 2011, pp. 27–57 (chapter 3).

[58] L. Najman, H. Talbot, Mathematical Morphology : from Theory to Applications, ISTE, London, 2010 Hoboken, NJ : Wiley.

[59] H. Irshad, A. Veillard, L. Roux, D. Racoceanu, Methods for nuclei detection, segmentation, and classification in digital histopathology: a review–current status and future potential, IEEE Reviews in Biomedical Engineering 7 (2014) 97–114.