Supplementary Information:

Materials and Methods

Supplementary Text

Figures S1 to S7

Tables S1-3

Materials and Methods:

1. ***Urine sample collection.*** The samples have been collected under the active IRB at the Dartmouth-Hitchcock Medical Center (DHMC) and Cleveland Clinic. Urine was collected in the middle of urination- avoiding first morning urination, following recommendations for VUC. 25 mL of urine was used to avoid inadequate cellularity, which is close to 30mL recommended in (*1*). The human subjects were recruited from healthy individuals, patients with previous history of bladder cancer who did not have cancer at the moment, and patients with bladder cancer. Diagnostic of cancer and the cancer stage were done using standard liquid cytology and cytoscopy protocols, and/or transurethral resection of tumor (TURBT).
2. ***Patients tested.*** Urine samples were collected from 22 patients identified with cancer (12 low- and 10 high- grade as defined by TURBT after collecting urine samples) and 43 non-cancer individuals (healthy volunteers and patients with previous history of cancer) at DHMC and 3 cancer (2 low and one high- grade) patients. The age of the subjects ranged from 54 to 87, about 20% females (this cancer is prevalent among males). The samples were processed as described above, fixed, and sent to the microscopy lab at Tufts University, where the cells were washed, freeze-dried, and processed through the three-step -protocol described below. As a result, cancer patients: 1 sample without cells (24 with cells), non-cancer individuals: 18 samples without cells (25 with cells).

3. ***Cell preparation protocol.*** The cell samples are extracted from urine samples by centrifugation. The precipitant is re-suspended in PBS buffer and fixed. The preliminary results were obtained with Karnovsky fixative (*2*). Following the fixation protocol (*3-5*), the cells are subsequently washed with DI water, and transferred to the AFM lab for analysis. To protect the cell from drying artifacts, the cells are dried using freeze-drying. To do that cells as received are first washed by centrifugation, re-suspension, and centrifugation again in clean DI water. The precipitant with a small amount of water is transferred on **a** precooled glass slide and quickly frozen using a standard freeze-dryer freezer (by Labconco) for five minutes. The glass slide with frozen sample is then placed in a freeze dryer operating in -45°C (the

time depends on the specific freeze dryer and the amount of sample; it can be as fast as 30 minutes). The AFM imaging is applied to cells or cell-like objects) randomly chosen with the help of an optical microscope build-in the AFM, see, e.g., Fig.1 A (main text). The cells are imaged directly on the glass slide taken from the freeze dryer. No further preparation of the sample is needed.

4. ***Cells chosen for the AFM imaging***. Cells prepared as described above, were ready for imaging with AFM. Specific objects to image were *chosen randomly* with the help of an optical microscope built-in the AFM setup. The only criterion was to pick up a relatively round object which looked like as a cell with the optical microscope.

   However, several objects, which we call "cell-like objects", were excluded from the analysis. The following protocol for the exclusion was used. Three representative optical images of these objects are demonstrated in Fig.S1 a-c. Optically these objects are hardly to distinguish from cells. However, AFM images of these objects showed an unusual layered structure. Figs.S1 d-g show representative examples of AFM images of these objects. One can see a distinctive layered structure, which had not been seen on biological cells (shown in Fig.1c-f). In addition, the adhesion channel showed clear horizontal lines on these objects, Fig.S1 e,g, which are very rarely seen on cells. These two features allowed us to unambiguously differentiate these objects and cells. These objects were not considered to be cells and not used for the analysis. As a result, the individual who had just such cell-like objects were classified as "no cells".

5. ***AFM imaging*.** Bioscope Catalyst (Bruker/Veeco, Inc., Santa Barbara, CA) atomic force microscope equipped with Nanoscope V controller was used to image cells found in urine. Bruker ScanAssyst cantilevers for imaging in air were used. To collect the maps of cells, two sub- resonant tapping modes were used, a standard PeakForce tapping (Bruker/Veeco, Inc., Santa Barbara, CA) , and new Ringing mode (NanoScience Solutions, Inc., Arlington, VA) (*6*). Both modes were verified to give the same surface parameters for the height and adhesion channels. The reason of using two modes was that ringing mode allows collecting images faster (initially data were collected using PeakForce tapping, and we switched to ringing mode later on). Although PeakForce tapping and Ringing modes allow collecting 6 and 14 different channels, respectively, only 2 channels were present in both modes and were sufficiently robust to be used for cell classification, cell height and adhesion. The images are collected at the scan size of 10x10 microns (at the resolution of 512x512 pixels). The speed of scanning is 0.1Hz in the PeakForce and 0.4Hz in ringing mode; the scan (peak) force is 5nN. Fig. 1 shows an example of images of bladder cells prepared as described above and imaged with the sub- resonant imaging (either PeakForce tapping or Ringing) mode.

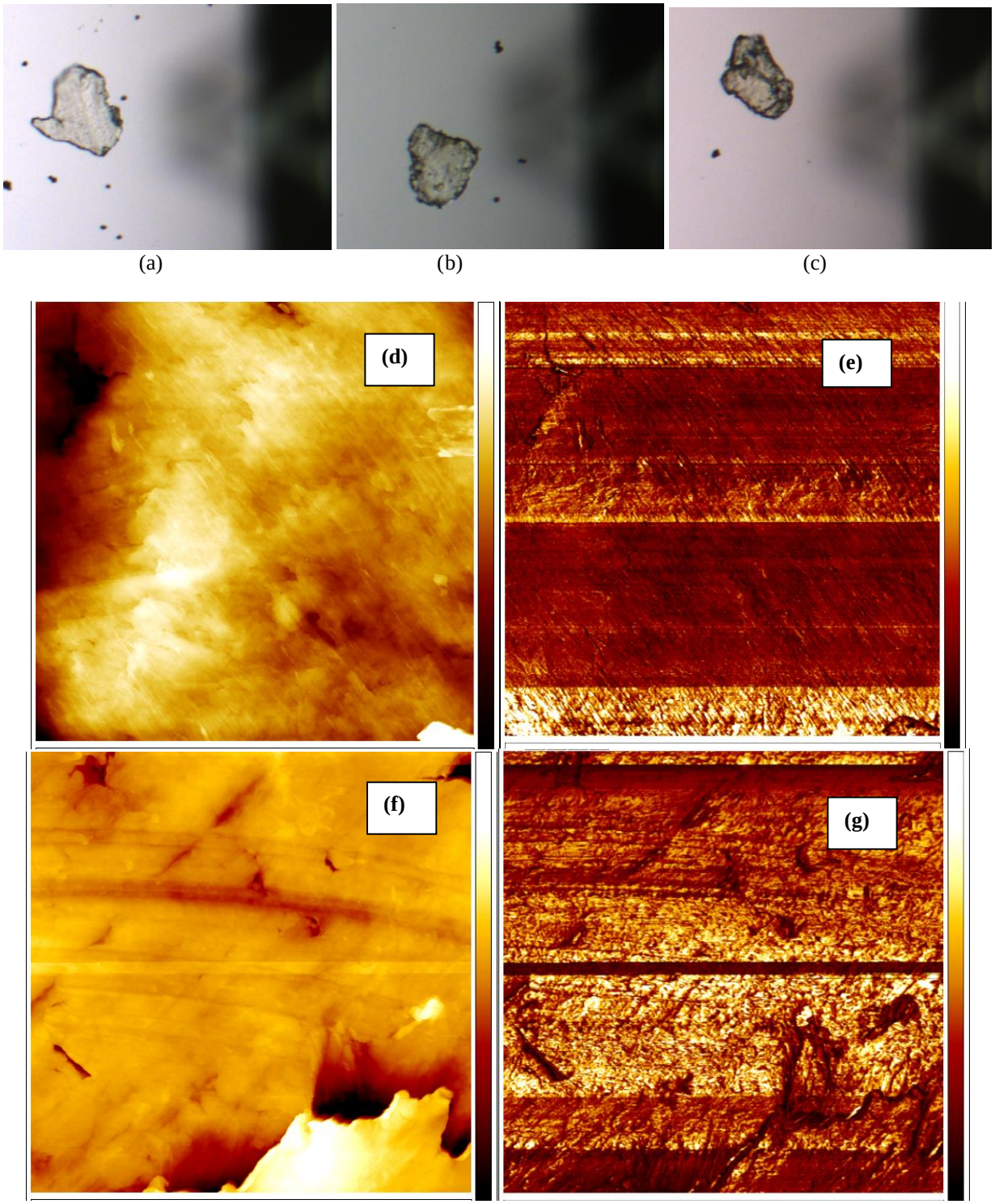


Figure S1. Representative examples of *cell-like objects*. (a-c) 400x600 μm^2 optical images of cell-like objects obtained using Nikon TE 2000U microscope; a shadow on the right comes from the AFM cantilever. (d-g) AFM recorded 10x10 μm^2 maps of height (d,f) in adhesion (e,g). *Horizontal lines* clearly seen in the adhesion channel are due to picking up debris by the AFM probe, a feature almost never seen on cells.

6. ***Calculation of surface parameters***. Surface parameters are routinely used in multiple engineering applications to characterize surfaces (*7*) We used software SPIP (by Image Metrology A/S, Denmark), which automatically calculates 44 surface parameters for 3D image surface arrays recorded by AFM. Here we apply it for both height and adhesion images (AFM images are digital arrays of either heights or adhesion, respectively). A complete list of surface parameters used in the present work is as follows: Roughness Average, Root Mean Square (RMS), Surface Skewness, Surface Kurtosis, Peak-Peak, Ten Point Height, Max Valley Depth, Max Peak Height, Mean Value, Mean Summit Curvature, Texture Index, Root Mean Square Gradient, Area Root Mean Square Slope, Surface Area Ratio, Projected Area , Surface Area, Surface Bearing Index, Core Fluid Retention Index, Valley Fluid Retention Index, Reduced Summit Height, Core Roughness Depth, Reduced Valley Depth, l-h% height intervals of Bearing Curve, Density of Summits, Texture Direction, Texture Direction Index, Dominant Radial Wave Length, Radial Wave Index, Mean Half Wavelength, Fractal Dimension, Correlation Length at 20%, Correlation Length at 37%, Texture Aspect Ratio at 20%, Texture Aspect Ratio at 37%. To take into account no-cells results, we added a new "no cell" parameter (to keep the same data structure assigned "no-cell" samples artificial negative values to the surface parameters; the statistical results do not depend on a particular value assigned).

Examples of a few surface parameters, which are among most important for the classification shown in Fig.2 of the main text, Svi, Sdr, S3A. They are defined as follows.

The Valley Fluid Retention Index ("Svi"):

$$Svi = \frac{V(h_{0.80})}{(M-1)(N-1)\delta x \delta y} / Sq \ , \qquad \text{(S1)}$$

where N and M are the number of pixels and x and y directions, $V(h_x)$, is the void area over the bearing area ratio curve and under the horizontal line h_x , S_q is the Root Mean Square (RMS) parameter defined as $Sq = \sqrt{\frac{1}{MN}\sum_{k=0}^{N-1}\sum_{l=0}^{M-1}\left[h(x_k, y_l)\right]^2}$. Large values of *Svi* indicate large void volumes in the valley zone.

Surfaces Area Ratio ("Sdr") parameter expresses the increment of the interfacial surface area relative to the area of the projected (flat) x, y plane:

$$S_{sd} = \frac{\left(\sum_{k=0}^{M-2}\sum_{l=0}^{N-2} A_{kl}\right) - (M-1)(N-1)\delta x \delta y}{(M-1)(N-1)\delta x \delta y} 100\% \ , \qquad \text{(S2)}$$

where N and M are the number of pixels and x and y directions, A_{kl} is defined as:

The Surface Area, ("S3A") is the 3D area of the surface given by the following formula:

$$S3A = \left(\sum_{k=0}^{M-2} \sum_{l=0}^{N-2} A_{kl} \right) - (M-1)(N-1)\delta x \delta y \,. \qquad (S3)$$

The surface parameters were calculated for the AFM images as follows. Each 10x10 μm^2 cell image of 512x512 pixels was split into 4 zoomed areas (5x5 μm2). Thus, each cell is quantified with 4 sets of surface parameters calculated for each quadrant. A few images (less than 134 out of 1,460) showed a clearly identified small round-shaped junk, see an example in Fig.S2. The images with that junk were excluded. Although it is possible to write an algorithm to identify this junk and make the method completely operator-independent, we noticed that these artifacts can easily be excluded by just considering median values of the parameters for each cell instead of the mean values. The results described in this work are virtually unchanged if we consider either median values per cell of all parameters without excluding the artifacts or average parameter values per cell excluding the artifacts.

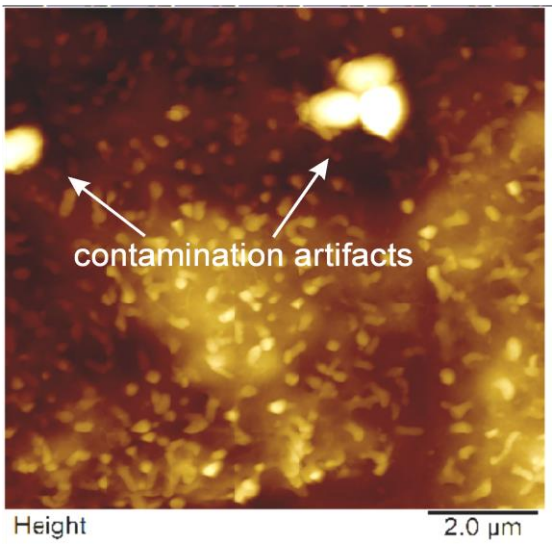


Fig.S2. A representative example of the artifacts because of possible contamination of the cell surface.

7. ***Use of human subjects*.** This study was approved by the Geisel School of Medicine Institutional Review Board of Dartmouth College, under trial registration number (CPHS Study: 29124), Cleveland clinic (IRB Case 2815; 14-1222 Comprehensive bladder cancer database), and Tufts University (Health Science IRB 12605).

8. ***Data availability statement*.** The data used in the work which support the findings of this study are available from the authors upon reasonable request and with permission of the Geisel School of Medicine. Restrictions apply to the availability of the medical

training/validation data, which were used with permission for the current study, and so are not publicly available.

9. ***Statistical analysis of the obtained results***. Here we analyze ROC (receiver operating characteristic) curve and the confusion matrix. ROC curve allows to define a range of sensitivity ("accuracy" of cancer diagnosis) and specificity ("accuracy" of healthy diagnosis), which are defined as follows:
sensitivity = TP/(TP+FN);
specificity = TN/(TN+FP);
accuracy= (TN+TP)/(TP+FN+TN+FP);

   It makes sense to define these two important parameters, sensitivity and specificity for a ROC point that corresponds to the *minimum error of classification* of both cancer and normal cases. It is shown in Tables 1 and Tables S3 and S4.

   In principle, the sensitivity and specificity can also be defined around so-called balanced point, in which sensitivity = specificity. Because of a limited number of human subjects, it is rather difficult to define the precise point when sensitivity = specificity, therefore, we defined this point as the one in which $|\text{sensitivity} - \text{specificity}| < 5\%$. These results are shown in Tables 1, S5 and S6.

10. ***Machine learning methods adopted to the present data structure***. To classify cells as coming from either cancer patients or non-cancer individual (two possible classes), and to check if the results stay similar for different machine learning methods, we chose three different methods: Random Forest, Extremely Randomized Forest, and Gradient Boosting Trees. These methods were chosen as the least prone to overtraining, a common problem of machine-learning methods. The first two methods are bootstrap unsupervised ones, and the last one is a supervised method of building trees. All data manipulation and analysis was carried out in Jupyter Notebook (version 4.2.1), which is an interactive Python desktop environment. We imported the data from SPIP (version 6.5) output using Pandas Python package (version 0.18.1). Variable ranking, classifier training and validations were calculated using appropriate Classifier functions from scikit-learn Python machine-learning package (version 0.17.1).

    Below we give a short description of these methods (detail description can be found in references (*8-15*)), highlighting the difference between these methods, and the adoption of our data structure to be used with these methods, as well as generalization of these methods to the case of diagnostics based on the analysis of multiple cells.

    Random Forest and Extremely Randomized Forest methods are based on growing many classification trees, so-called bootstrap methods. Each of such trees predicts some classification, whereas the final classification is defined by votes of all trees of the method. The trees are grown on the training part of the total data set. It is typical to use 70% of all data for training/growing the tree, and us the remaining 30% for validation of the method, testing the accuracy of the training. These splitting are random, and repeated multiple times (similar to the

known Monte-Carlo idea). If there are N_p input variables (the surface parameters in our case), a subset of these variables is randomly chosen out of N_p input variables for each branching node (we use the default, a squire root of N_p). The best split of the tree branches with these chosen parameters is found (each node is a split on one of the chosen parameters). The criteria for the split threshold is based on estimation of the classification error. The classification error criterion for binary splits cannot be defined as a usual statistical error of measurements. It has been suggested to use the classification error rate. Specifically, each parameter is assigned to a parameter region with respect to the most commonly occurring class of the training set. The classification error is defined as a fraction of the training samples in that region that does not belong to the most common class:

$$E = 1 - \max_k(p_{mk}) \quad . \tag{1}$$

Here p_{mk} represents a proportion of training samples in the mth region that belongs to the kth class. However, for a practical use, equation (1) is not sufficiently sensitive to unnecessary overgrowing the tree. Thus two other measures have been introduced, the Gini index and cross-entropy. The Gini index is defined as follows:

$$G = \sum_{k=1}^{K} p_{mk}(1 - p_{mk}) \quad . \tag{2}$$

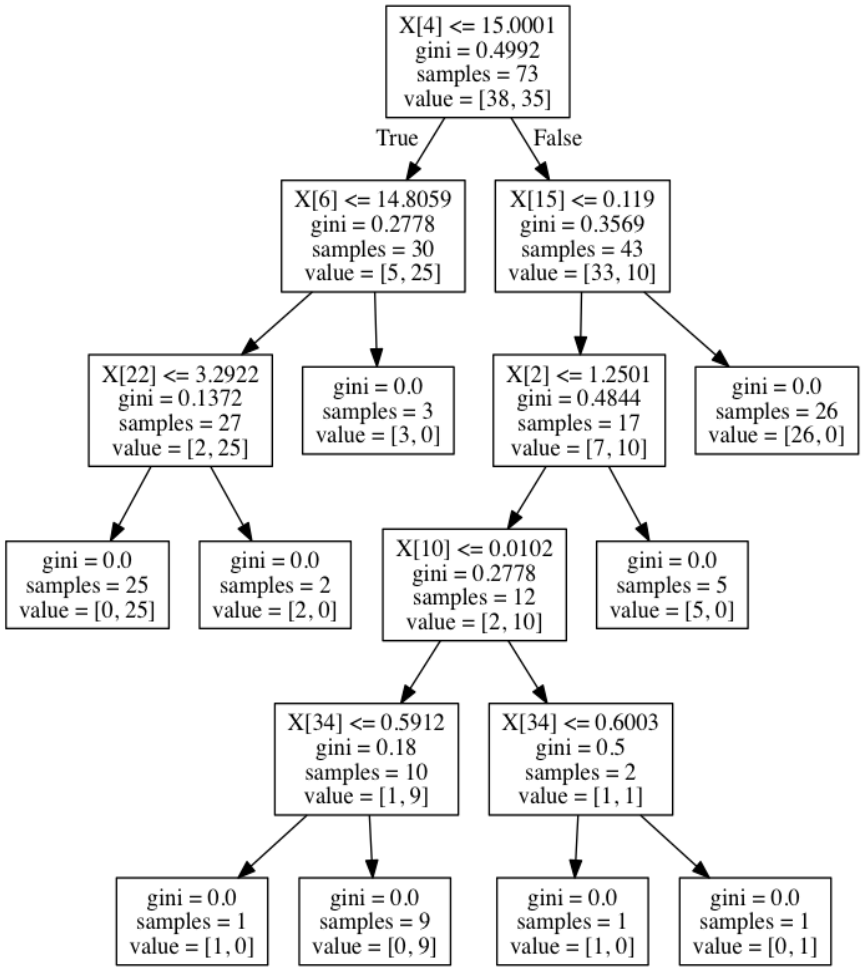
It represents a measure of total variance across all K classes. The Gini index has a small value if all p_{mk}s are close to zero or one, which means that it can be used to measure node purity (meaning that a particular node contains mostly samples from a single class). For example, each tree is grown until the Gini-index results in complete separation of classes (Gini impurity criterion: for the two descendent nodes the Gini-index is less than the one for the parent node). There is no pruning of the growing brunches in these Random Forest methods.

The cross-entropy is defined as:

$$D = -\sum_{k=1}^{K} p_{mk}\log(p_{mk}) \quad . \tag{3}$$

Similarly to the Gini index, cross-entropy has a small value if all p_{mk}s are close to zero, and it is indicative of a pure node. To obtain a measure of the importance of each parameters (importance coefficient or variable importance), one can use, for example, the Gini index. We add up all values of the decrease of the Gini index at the tree nodes for each of the variables, and average over all the trees (*16*). A importance coefficient plot can be made in the form of a histogram shown in Figs.2, a,b and S7.The number of trees used was 100-300 (there was no noticeable dependence of the results on the number of trees), the maximum number of variables used was the square root of the total number of variables available (default), and classification error criterion was the Gini index. An example of a binary tree build as described above is shown in Scheme S1.

The described above method is implemented for in our bootstrap methods. Extremely randomized trees method is different from Random Forest in terms of the choice of the split. Instead of computing *optimal* parameter/split combination (using Gini index) for Random Forest, each parameter value is randomly selected from the parameter empirical range. To keep such a random choice converge to the pure classification (zero Gini index), only the best split among random uniform splits in the set of selected variables for the current tree is chosen.



Scheme S1. One of the trees from the ensemble out of 100-300 trees used in our bootstrap methods. In the first split, the fourth variable was chosen with split value of 15.0001, which yielded the Gini index of 0.4992 and split 73 samples (38 class 1 and 35 class 2) into two bins, each having 30 and 43 samples, respectively. At the second level split, looking at left hand side node, the sixth variable was chosen with split value of 14.8059, which yielded the Gini index of 0.2778 and split 30 samples (5 class 1 and 25 class 2) into two bins with 27 and 3 samples, respectively. The split continues until a tree node has the Gini index of 0 indicating presence of only one of the two classes.

Gradient Boosting Trees is a technique based on the building of a series of trees, each of which converges with respect to some cost function. Each subsequent tree is built to minimize the deviation from the exact prediction (for example, the mean squared error). The (Friedman)

algorithm of "treeboost" is used in this process of regression (implemented in the use scikit-learn Python package). Because there is no criterion for pure nodes (like the one based on Gini index in the previous two methods), the size of the tree has to be predefined, or alternatively, by the limiting of the number of individual regressions (maximum depth). The trees built in this way can easily be overfitted though. To avoid the overfitting effect, there are some constraints imposed, like the number of boosting iterations and weakening the iteration rate (done by introducing a dimensionless parameter, which is called the learning rate). It is also possible to limit the minimum number of trees terminal nodes (minimum number of leaves). In the used here algorithms (scikit-learn Python package) the following parameters were used: minimum number of leaves=1 (default), learning rate = 0.01, the maximum depth=3 (default). Note that the learning rate was put 10x less than the default value. This was chosen to decrease variance due to a relatively small finite number of human subjects. Other multiple parameters were used at their default values. The only exception was in the use of subsample parameter (0.45 instead of 1). This parameter dictates the use of the stochastic gradient boosting. It was chosen to reduce variance (frequently at the expense of bias), which might be large due to a relatively limited number of human subjects.

Specific care was used to create the appropriate data sets for training and verification. The algorithm is described in Scheme S2. In addition, we deal with a hierarchical data structure: each human subject has several cells, and each cell has several (four) images associated with it. Each image was evaluated visually for the presence of artifacts shown in figure S2. If an artifact was present, the corresponding data set was assigned the attribute of artifact. These data were ignored in the later analysis. Because for different areas were analyzed per each cell, we first combine them for each cell by either averaging or taking median (no significant difference in using either one was observed; the results presented in this work are done for the median values of the parameters calculated for each cell). Then, the data set was randomly split to have S% of the total data for training and 100-S% for testing. Here we consider S=50, 60, 70%. The splitting is done by keeping the data from the same individuals in just one of the either subsets to avoid artificial over-training due to correlation between different cells of the same individual.

To decrease the number of variables/surface parameters, the following algorithm was used. First, we rank the parameters with respect to the Gini index (importance coefficient as defined above), for example, Fig. 2a,b. Then, we keep N_p best parameters for each channel (height and adhesion), which are identified by (i) their segregation power and (ii) low inter-parameter correlation (to exclude the parameters that are correlated with the others, see Section 1 of the supplementary materials below). The dependence of the classification accuracies on the number of used parameters are shown in figure 2c,f.

To test the verification data set (or to test any new test data), the trained trees have to be used to obtain the answer on the predicted class. One can get the results of tree "voting", and consider it as a "probability of prediction". If the probability of belonging to class A greater than a threshold (default is 0.5), the result belongs to class A. When building ROC curve, the threshold should be moved.

Scheme S2.

Algorithm Cancer Detection using Cell Surface Parameters

function make_dataset() **Inputs:** CSV file, statistics parameter [mean or median]

Steps:

1) Load data from CSV file
2) Reject variables where all samples have the same value
3) Reject samples where data acquisition artifacts are present
4) Store data-frame with preprocessed data
5) Create data-frame with non-cancer samples with no cells
6) Merge preprocessed and non-cancer sample data-frames
7) Group samples by patient ID
8) Get mean or median values of all variables per cell, from cell quadrants, for all patient IDs and store in data-frame

Return: Cell samples data-frame

function split_data_train_test() **Inputs:** data-frame, train-test sample ratio

Steps:

1) Split data-frame into training and testing samples, based on patient IDs, using selectable train-test sample ratio
2) Store training and testing samples data-frames

Return: training and testing samples data-frames

function make_classifier_model() **Inputs:** model name, model parameters

Steps:

1) Select one of the following classification models: random forest, extra-trees, or gradient boosted trees
2) Specify selected model's parameters and instantiate the model

Return: Classification model object

function classification_performance_metrics() **Inputs:** classification model object, cell samples data-frame, M number of cells, N number of cells, R number of iterations

Steps:

for *n-th run in R number of iterations:*

run ***make_classifier_model()***

run ***split_data_train_test()***

1) Select M number of cells per patient ID to be used in measuring classification performance metrics for testing data-frame
2) Select N number of cells per patient ID which need to be classified as cancer to classify the whole patient ID as cancer
3) Using all cells available per patient ID, make all possible combinations of M cells for all patient IDs in the testing samples and store in data-frame
4) Calculate model classification performance metrics: ROC curve, ROC AUC score, accuracy, and confusion matrix

Return: ROC curves, ROC AUC scores, accuracy scores, confusion matrices

Supplementary Text

1. *Cross-correlation between surface parameters*.

Historically several of the surface parameters (their total number is 44) are just redundant redefinition of others. Here we excluded already several of these parameters because of explicit 100% correlation with other parameters (based on their formal definition). It reduces the number parameters to 39. Next, two parameters (Srw and S2A) were dropped as ill-defined for our particular type of surfaces, which leaves us with only 37 surface parameters. Further reduction is done through the analysis of correlation coefficients between these parameters. Obviously, the parameters having high correlation between each other carry redundant information, and therefore, can be removed with a little penalty to the classification power.

To find the correlation matrix between surface parameters, we analyzed specific formulas defining some parameters and calculate the parameters for a large family of random surfaces. The latter was done due to the complexity of definition of some parameters, which were defined by an algorithm rather than a single formula. The random surfaces were generated with the help of iPython language using the algorithms described in [1]. An example of a simulated surface is shown in figure S3. One can see a good resemblance to realistic services.

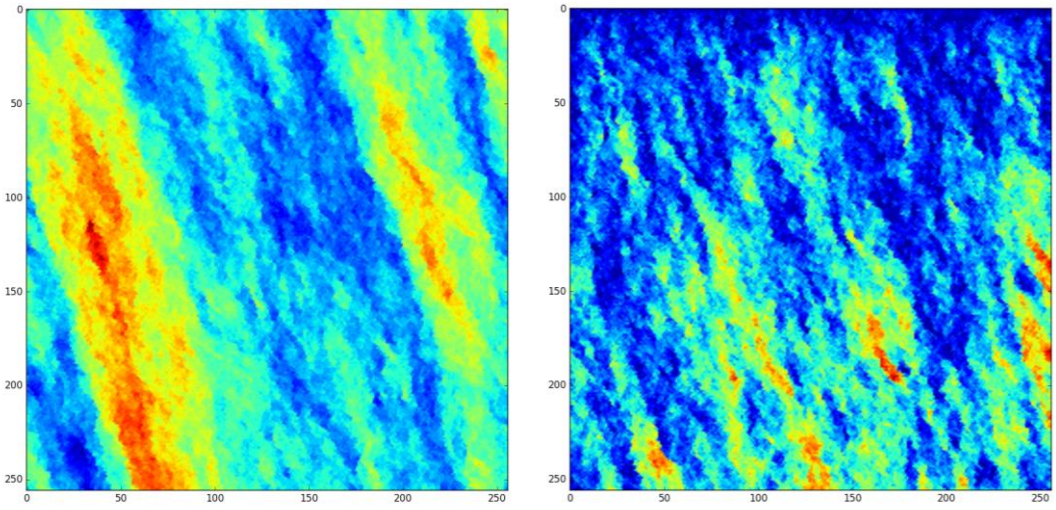


Figure S3. Two examples of a simulated surface used to estimate correlation between different surface parameters.

Various surfaces of cells imaged with AFM were also used to test the parameters correlation. The surface parameters were calculated by using SPIP software. The correlation matrix obtained based on about hundred cells and 7000 similar to surfaces. The convergence analysis of the coefficients of correlation matrix showed good convergence after using approximately 1000 simulated surfaces, see figure S3 as an example. The correlation matrix is shown in Supplementary Table 1.

See the attached corr.csv file

Table **S1**. Values of the correlation coefficients between the surface parameters.

Considering different threshold for the correlation parameter, one can find the number of surface parameters selected. If the threshold is zero, we have just one parameter. If it is 1, then we have all surface parameters chosen. Figure S4 shows the dependence of the number of surface parameters selected on the different threshold of the correlation parameter for the example of Random Forest model. The other models show similar dependences.

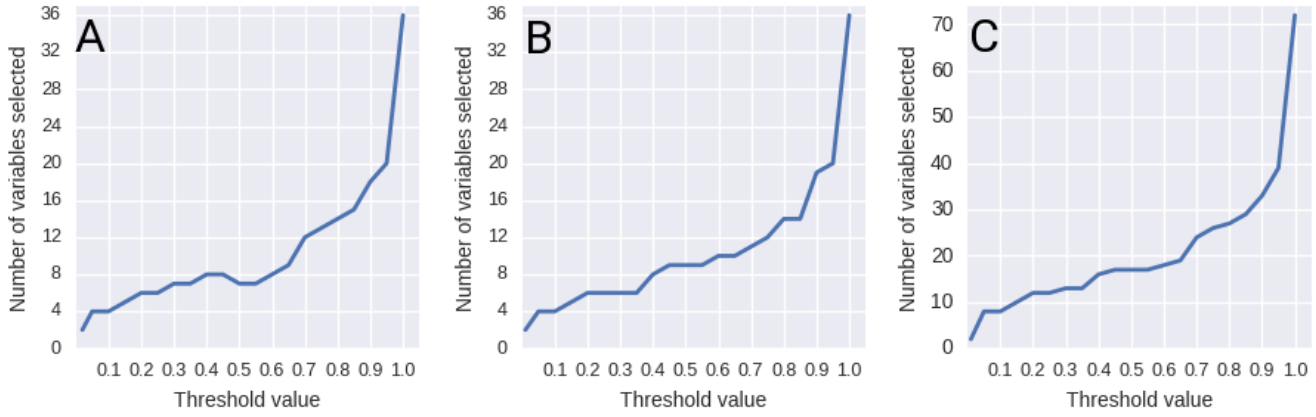


Figure S4. RANDOM FORESTS. Number of variables selected per threshold value by comparing correlation coefficients of the most important variable and the reminder of variables for: (A) height channel; (B) adhesion channel; (C) combined height and adhesion channels.

2. Why the described method can easily tolerate presence of cells carrying no-cancer signature in samples from cancer patients

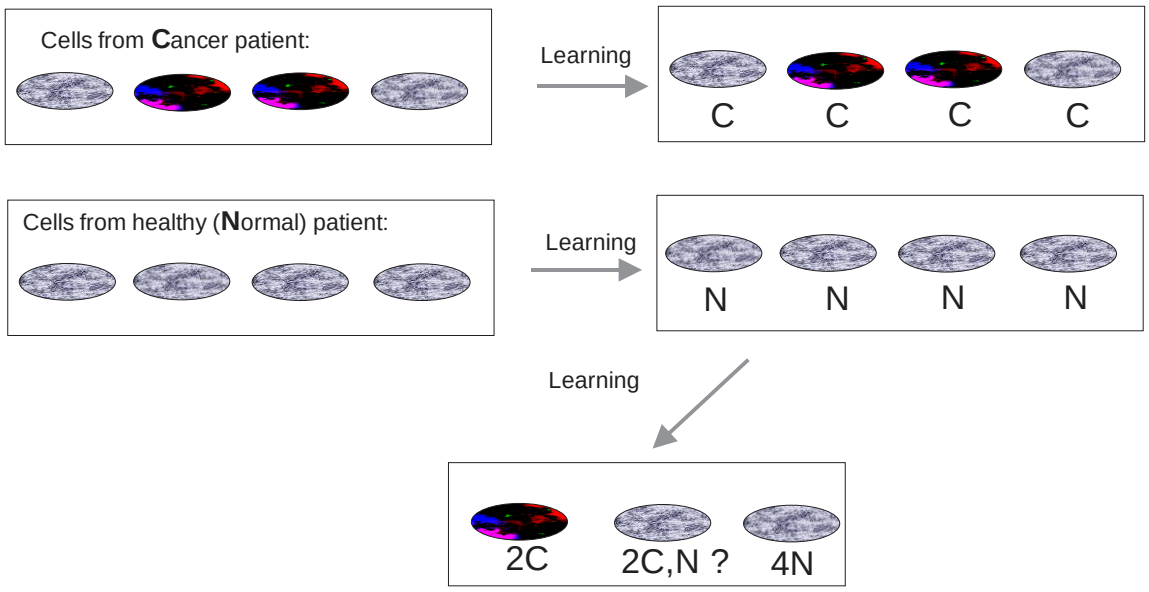
Based on our results, we definitely have a confirmation of the field carcinogenesis approach. Consequently, the signature of cancer should be all cells exfoliated from the cancerous bladder. However, it is known that a number of cells can be exfoliated (squamous) not from bladder but from the rest of the urinal tract. It is no clear reason why the squamous cells should care any signature of bladder cancer. However, *here we demonstrate that the described method has a high tolerance to the presence of such cells, and the accuracy of cancer detection can still be very high.*
Before doing that, let us note about other than squamous cells mentioned above. Besides urothelial cells coming from bladder, urine can also have red blood cells, white blood cells, neutrophils.

However, such cells can easily be separated with the help of an optical microscope based on their small size. Thus, the only confusion one may have is a mix of squamous and urothelial cells. Although urothelial cells are typically a bit smaller than squamous cells, it is not easy, and sometimes, virtually impossible to distinguish between these two types of cells. So, it is natural to assume that our sample will contain some amount of squamous cells.
Let X% of cells extracted from urine of a cancer patient carry a signature of cancer. It is natural to assume that the number of such cells would be more than number of cells exfoliated from other places of the body, i.e., X>50%. Nonetheless, let's start from an example of the algorithm work for the case of X=50% (only half of the cells extracted from urine of cancer patients carry the signature of cancer ("cancerous" cells).

The algorithm works as follows:

A. **The learning stage**.
   Assume that we image 4 cells for each patient. Then, the most probably one gets 2 cancer cells for a cancer patient and 0 for normal. Because our algorithm "learns" the features of cancer and normal cells, after learning from one cancer and one normal samples, it will identify (see **Fig.S5**):
   - 2 distinctive cells (derived and defined as cells coming from cancer patient),
   - 2 ubiquitous cells (derived and defined as cells coming from both cancer and normal patients),
   - 4 distinctive cells (derived and defined as cells coming from normal patient).

   For the sake of simplicity, let us assume that our algorithm can have 100% accuracy to segregate between cancer (C) and non-cancer (N) cells. Thus, assuming random choice of a cell, one can identify one true cancer cell with accuracy of 50%, and normal cells with accuracy of 67% (4 normal cells from normal patient /6 all normal cells). This is certainly an oversimplified calculation for demonstration only. We considered only the most probable situation. In reality even with X= 50%, there are probabilities of having from 0 to 4 cancer-signature carrying cells in the cancer patient sample; all that have to be taken into account to calculate the exact accuracy. A particularly "dangerous" case of cancer-signature carrying cells is considered below; its probability is estimated.



where

is a cell that carries some signature of cancer (C).

Is a cell that carries **no** signature of cancer (N).

Fig.S5. An example of work will be used algorithm for the case of testing 4 cells, when X=50% (only half of the cells extracted from a cancer patient carries some signature of cancer).

B. **Testing stage**.

One can define diagnosis of cancer if at least M cells are identified by the algorithm as cancerous (derived from the cancer patients). ***Within the above example, if M=1,2 the accuracy of finding cancer will be 100%.*** *If M=3,4, one will start missing cancer patients*. In reality the accuracy is definitely lower because the algorithm is not 100% accurate.

Note about possible complete missing of cancerous cells in cancer patient samples

Let us estimate what is the probability of missing a cancer cell when we randomly pick cells for AFM imaging. If one considers a rather conservative percentage of cancer-signature carrying cells in the entire urine sample, X=50%, the probability of not having cancer cells in the sample of N cells is $(1/2)^N$. For example if N=4, it is 6%. This can be neglected if you are speaking about accuracy to identify cancer ~90%.

In reality, the probability of missing cancer cells is even less because X>50% (the percentage of cells in the urine sample which are exfoliated from bladder).

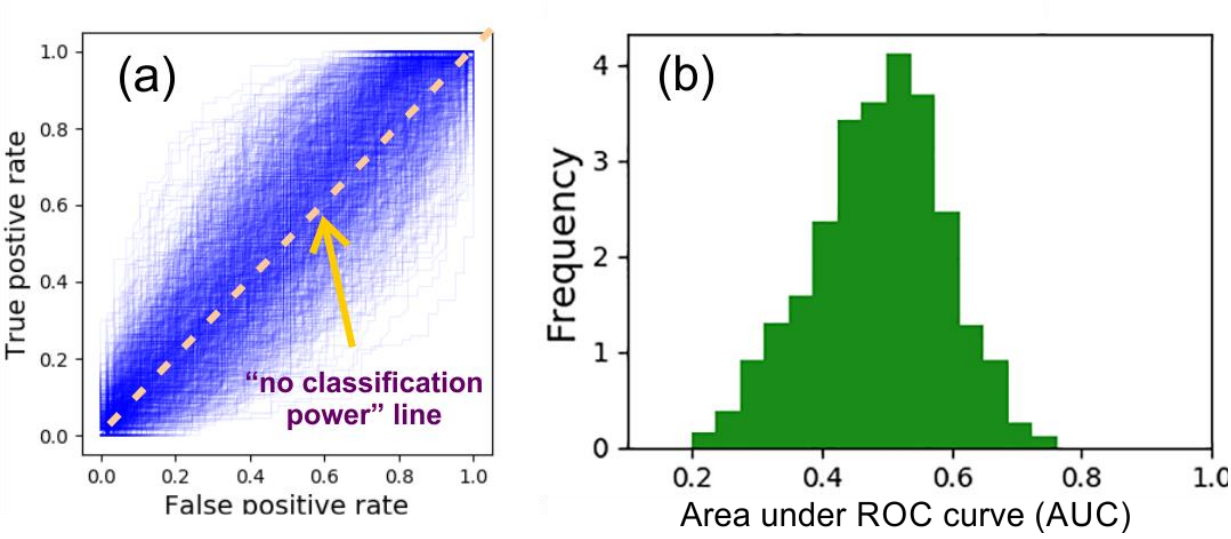


Fig.S6. Absence of overtraining. Example of Random Forest Method. (a) ROC curves are shown for the same algorithms and data as in Fig.2 but with ***randomized diagnoses***. One can see no "cancer" detection (which could be if the results were an artifact of overtraining). (b) histogram of AUC showing the scatter around 0.5 (no classification power).

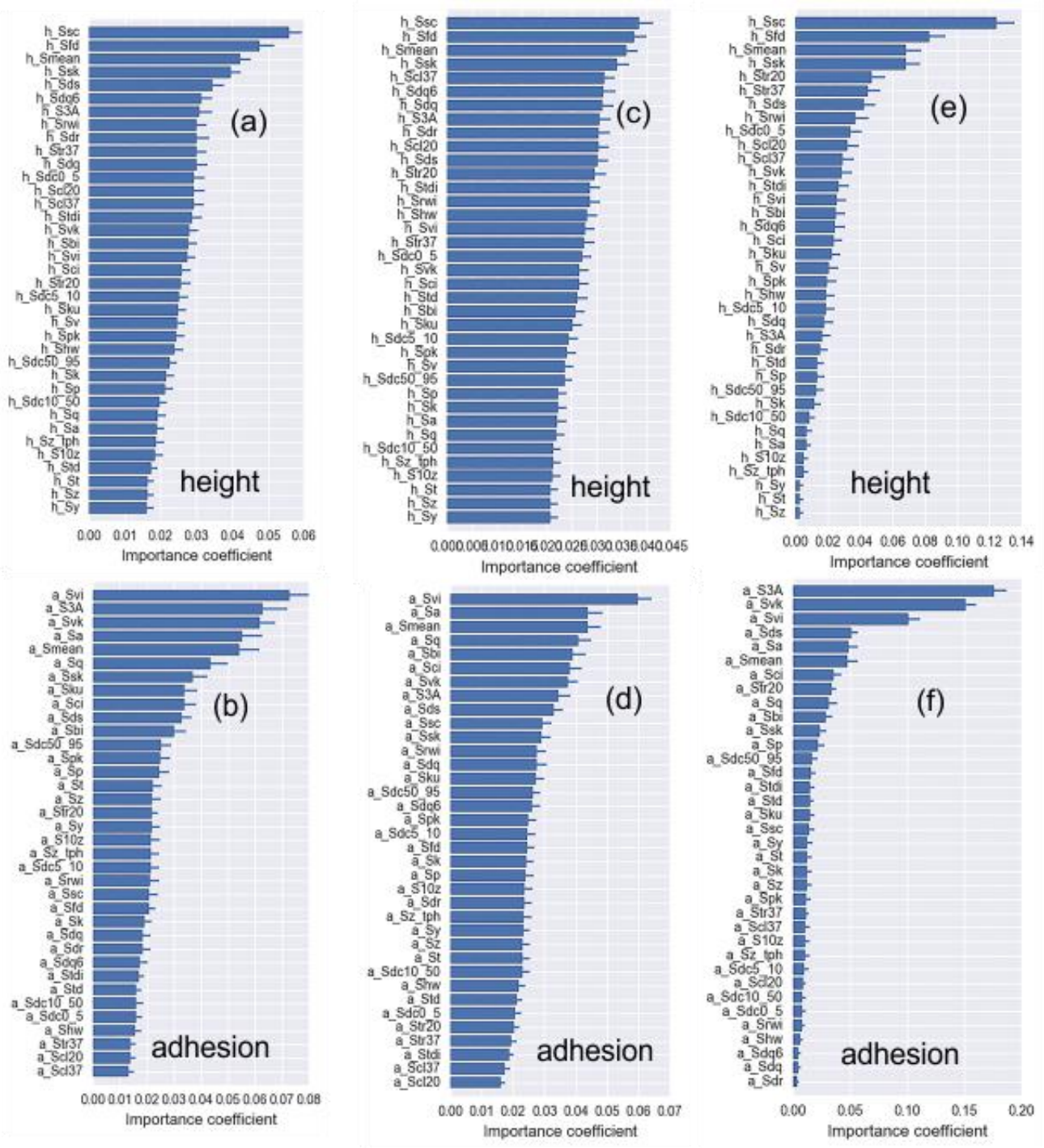


Figure S7. The importance coefficient indicating segregation power between normal and cancer patients (Gini-index measure) of the surface parameters for (a) height, (b) adhesion. The averaged value and one standard deviation of each parameter are shown for the Random Forest method (a,b), Extremely Randomized Forest (c,d), and Gradient Boosting Trees (e,f). 1000 randomly chosen (70/30%) splits for training and testing data sets were used.

Table S2. Statistics of diagnosis of cancer for an individual by considering N cells and requesting that M cells out of the considered N ($M \leq N$) were classified as collected from a cancer patient to put diagnosis of cancer. Sensitivity and specificity, averaged AUC and accuracy were calculated for 1000 random splits of the entire data onto training and versification sets (70% training and 30% verification split) for all three methods. The accuracy is found for the smallest error of classification. Sensitivity and specificity are given for that case (the left colon of Sens/Spec part of the table); the right colon of Sens/Spec part is another example demonstrating higher sensitivity (the threshold to separate cancer from noncancer cases was chosen to keep the difference between sensitivity and specificity close to 5%). The best case is highlighted. *All data was used*.

	data	Random forest			Extremely Randomized Forest			Gradient Boosting Trees		
		AUC/ Accuracy	Sens/spec		AUC/ Accuracy	Sens/spec		AUC/ Accuracy	Sens/spec	
N=1	height	75/73	50/84	76/69	77/74	53/84	77/70	74/73	46/86	75/68
M=1	adh.	88/83	68/90	84/77	88/83	69/90	85/78	87/82	69/89	84/77
N=2	height	75/77	40/91	76/69	77/78	44/91	77/71	75/77	40/92	75/69
M=1	adh.	89/87	71/93	86/80	89/86	70/93	87/80	89/87	71/93	86/80
N=2	height	77/78	43/92	76/69	78/79	48/91	78/71	75/78	38/93	74/68
M=2	adh.	89/86	69/93	86/80	90/87	70/94	86/79	89/87	69/94	85/79
N=3	height	73/81	34/95	75/66	74/81	35/95	76/67	73/81	35/95	74/66
M=1	adh.	90/89	70/95	88/81	89/88	69/94	86/79	89/90	70/96	87/80
N=3	height	75/82	36/95	75/68	78/82	42/94	77/69	73/81	34/96	73/66
M=2	adh.	91/90	73/96	87/80	90/90	72/96	87/80	89/90	69/96	85/78
N=3	height	75/82	37/95	76/67	78/83	42/95	77/69	72/81	32/96	73/66
M=3	adh.	90/90	71/96	87/80	90/90	70/96	87/79	88/89	65/96	85/77
N=4	height	69/84	31/97	71/61	70/84	33/97	73/64	69/84	34/97	73/64
M=1	adh.	89/91	75/95	88/80	88/90	71/96	87/79	90/92	74/96	88/80
N=4	height	72/84	35/97	73/64	74/85	37/96	76/67	71/84	37/96	72/63
M=2	adh.	91/92	77/97	88/80	90/92	74/96	88/80	90/92	73/97	87/79
N=4	height	72/84	34/97	74/64	76/85	41/96	76/67	70/84	32/97	71/62
M=3	adh.	89/92	72/97	88/79	90/92	72/97	87/79	87/91	64/97	84/75
N=4	height	72/84	32/97	74/64	75/85	38/96	76/66	70/84	32/97	72/62
M=4	adh.	89/91	71/96	88/79	89/91	71/96	87/78	86/90	64/97	84/75
N=5	height	66/86	33/98	69/57	66/85	36/97	72/60	67/86	35/98	70/59
M=1	adh.	88/93	77/96	87/78	87/92	75/96	87/78	89/94	77/97	88/79
N=5	height	69/86	34/97	72/61	70/86	39/96	76/64	68/86	36/97	70/61
M=2	adh.	91/94	81/98	91/82	90/94	78/97	89/80	91/94	78/98	88/80
N=5	height	68/86	31/98	72/61	72/87	39/97	76/65	67/86	34/97	69/58
M=3	adh.	90/93	77/97	88/78	90/93	77/97	88/78	88/93	70/98	85/75

N=5	height	67/86	34/97	72/60	72/87	39/97	76/64	65/85	30/97	69/57
M=4	adh.	89/93	75/97	87/77	88/93	73/97	88/77	85/92	61/98	84/74
N=5	height	68/86	30/97	72/59	74/87	39/97	76/63	67/85	32/97	71/59
M=5	adh.	88/93	74/97	88/77	88/93	73/97	88/77	84/91	62/97	84/73

Table S3. Statistics of diagnosis of cancer for an individual by considering *N* cells and requesting that *M* cells out of the considered *N* ($M \leq N$) were classified as collected from a cancer patient to put diagnosis of cancer. Sensitivity and specificity, averaged AUC and accuracy were calculated for 1000 random splits of the entire data onto training and versification sets (70% training and 30% verification split) for all three methods. The accuracy is found for the smallest error of classification. Sensitivity and specificity are given for that case (the left colon of Sens/Spec part of the table); the right colon of Sens/Spec part is another example demonstrating higher sensitivity (the threshold to separate cancer from noncancer cases was chosen to keep the difference between sensitivity and specificity close to 5%). The best case is highlighted. *Only data with cells was used*.

	data	**Random forest**			**Extremely Randomized Forest**			**Gradient Boosting Trees**		
		AUC/ Accuracy	**Sens/spec**		**AUC/ Accuracy**	**Sens/spec**		**AUC/ Accuracy**	**Sens/spec**	
N=1	height	60/64	35/84	62/55	62/65	40/82	63/56	60/65	35/85	62/54
M=1	adh.	80/77	63/87	76/68	80/77	65/86	76/69	80/77	64/87	76/68
N=2	height	62/71	31/92	63/56	63/72	33/91	65/58	62/72	34/91	63/56
M=1	adh.	84/83	68/91	79/72	82/82	65/90	78/72	82/83	66/91	79/73
N=2	height	63/72	33/91	64/57	65/73	38/90	66/59	60/71	31/91	62/56
M=2	adh.	83/83	66/92	78/72	83/83	65/93	78/72	83/83	67/92	78/72
N=3	height	62/78	28/96	63/55	64/78	29/95	65/57	63/78	30/96	65/57
M=1	adh.	85/88	69/95	82/75	83/86	65/94	80/73	85/88	69/95	82/74
N=3	height	63/78	29/95	65/57	67/79	35/95	68/60	62/78	28/96	63/56
M=2	adh.	86/88	70/95	82/74	85/87	69/94	81/73	86/88	70/96	82/75
N=3	height	63/77	30/95	65/57	66/78	36/94	67/59	60/77	26/95	63/55
M=3	adh.	84/87	68/95	82/74	84/87	66/95	80/72	84/87	67/95	81/73
N=4	height	62/83	30/97	64/54	63/82	28/97	65/56	64/83	35/97	66/57
M=1	adh.	86/91	75/95	84/75	85/90	70/95	81/72	87/92	67/95	84/76
N=4	height	63/82	29/97	65/57	66/83	36/96	69/60	63/82	30/97	64/56
M=2	adh.	87/92	76/97	85/77	86/91	73/96	84/76	87/92	74/97	85/77
N=4	height	61/82	28/97	66/57	66/82	34/96	69/59	59/81	25/97	62/53
M=3	adh.	86/91	72/96	84/75	86/90	70/96	83/74	86/91	71/97	83/75
N=4	height	62/82	28/97	66/56	66/84	35/96	69/59	59/81	24/97	62/52
M=4	adh.	85/90	71/96	82/73	84/90	70/96	83/73	84/90	69/96	83/74
N=5	height	61/86	31/98	66/53	63/86	32/98	66/54	63/86	34/97	67/56

M=1	adh.	86/93	78/96	86/75	85/92	74/96	84/74	87/94	78/97	85/75
N=5	height	62/85	31/97	66/56	65/86	34/97	69/59	64/86	34/97	67/58
M=2	adh.	90/95	82/98	88/79	89/94	79/97	86/77	90/95	80/98	87/79
N=5	height	60/85	28/98	65/55	65/86	37/97	68/57	58/84	29/97	62/52
M=3	adh.	88/94	80/97	87/77	86/92	73/97	85/75	87/93	75/98	86/76
N=5	height	61/85	29/97	66/54	65/86	35/97	69/57	56/85	25/98	62/51
M=4	adh.	86/93	76/97	86/75	85/92	73/97	85/74	85/93	73/97	85/74
N=5	height	60/85	24/98	67/54	66/86	33/97	70/57	58/85	24/98	62/50
M=5	adh.	84/93	75/97	86/75	85/92	71/96	84/72	85/93	73/97	85/74

Supplementary references

1. C. J. VandenBussche, D. L. Rosenthal, M. T. Olson, *Cancer Cytopathol* **124**, 174-180 (2016).
2. M. E. Dokukin, N. V. Guz, R. M. Gaikwad, C. D. Woodworth, I. Sokolov, *Physical Review Letters* **107**, 028101 (2011).
3. I. Sokolov, *Future Oncology* **11**, 3049-3051 (2015).
4. N. V. Guz, M. E. Dokukin, C. D. Woodworth, A. Cardin, I. Sokolov, *Nanomedicine* **11**, 1667-1675 (2015).
5. M. E. Dokukin, N. V. Guz, C. D. Woodworth, I. Sokolov, *New J Phys* **17**, 033019 (2015).
6. M. E. Dokukin, I. Sokolov, *Scientific reports* **7**, 11828 (2017).
7. The complete list of these parameters is described in standards ISO 4287/1 ASME B46.1; ISO/DIS 25178-2 used to characterize surfaces in material science and engineering.
8. M. Sariyar, A. Borg, K. Pommerening, *J Biomed Inform* **45**, 893-900 (2012).
9. S. A. Azer, A. G. Frauman, *Ann Acad Med Singapore* **37**, 204-209 (2008).
10. D. K. Dunn-Walters, H. Edelman, R. Mehr, *Biosystems* **76**, 141-155 (2004).
11. P. Kokol, M. Zorman, M. M. Stiglic, I. Maleiae, *Stud Health Technol Inform* **52 Pt 1**, 529-533 (1998).
12. M. Zorman, M. M. Stiglic, P. Kokol, I. Malcic, *J Med Syst* **21**, 403-415 (1997).
13. J. Zavrsnik *et al.*, *Medinfo* **8 Pt 2**, 1688 (1995).
14. P. Kokol, M. Mernik, J. Zavrsnik, K. Kancler, I. Malcic, *J Med Syst* **18**, 201-206 (1994).
15. F. M. Wolf, *Acad Med* **68**, 542-544 (1993).
16. B. McMurray, G. Hollich, *Dev Sci* **12**, 365-368 (2009).