

Automated detection of bacterial growth on 96-well plates for high-throughput drug susceptibility testing of *Mycobacterium tuberculosis* – Supplemental Information.

Philip W. Fowler^{*1,2}, Ana Luíza Gibertoni Cruz¹, Sarah J. Hoosdally¹, Lisa Jarrett³, Emanuele Borroni⁴, Matteo Chiacchiaretta⁴, Priti Rathod³, Sarah Lehmann⁵, Nicolay Molodtsov⁵, Timothy M. Walker^{1,2}, Esther Robinson³, Harald Hoffmann⁵, Timothy E. A. Peto^{1,6}, Daniela Maria Cirillo⁴, E. Grace Smith³, and Derrick W. Crook^{1,6}

¹*Nuffield Department of Medicine, John Radcliffe Hospital, University of Oxford, Headley Way, Oxford, OX3 9DU, UK*

²*National Institute for Health Research (NIHR) Oxford Biomedical Research Centre, John Radcliffe Hospital, Headley Way, Oxford, OX3 9DU, UK*

³*National Mycobacterial Reference Service, National Infection Service, Public Health Laboratory Birmingham, Heartlands Hospital, Bordesley Green, Birmingham, B9 5SS, UK*

⁴*Emerging Bacterial Pathogens Unit, Division of Immunology, Transplantation and Infectious Diseases, IRCCS San Raffaele Scientific Institute, Milan, Italy*

⁵*SYNLAB Gauting and Institute of Microbiology and Laboratory Medicine, World Health Organization Supranational Reference Laboratory of Tuberculosis, Munich-Gauting, Germany*

⁶*NIHR Health Protection Research Unit in Healthcare Associated Infection and Antimicrobial Resistance at University of Oxford in partnership with Public Health England, Oxford, UK*

^{*}To whom correspondence should be addressed: philip.fowler@ndm.ox.ac.uk

List of Figures

S1	Raw images of fifteen plates from the testset.	3
S2	The testset after a mean shift filter has been applied	7
S3	A contrast limited adaptive histogram equalization improves contrast.	8
S4	Stretching the pixel histogram to ensure a standard brightness.	9
S5	An example of when AMyGDA failed to detect 96 wells	10
S6	The effect of varying the size of the central measuring region	13
S7	The cumulative pixel histograms for all wells in the testset images	14
S8	Performance of the binary classification process.	15
S9	Choosing parameters to optimise the sensitivity and specificity	16
S10	Avoiding artefacts by making the growth threshold a function of the control well growth	17
S11	The software produces an annotated composite image with all detected growth labelled	19
S12	The testset with wells containing growth highlighted.	20
S13	Example images showing the reproducibility of AMyGDA.	21

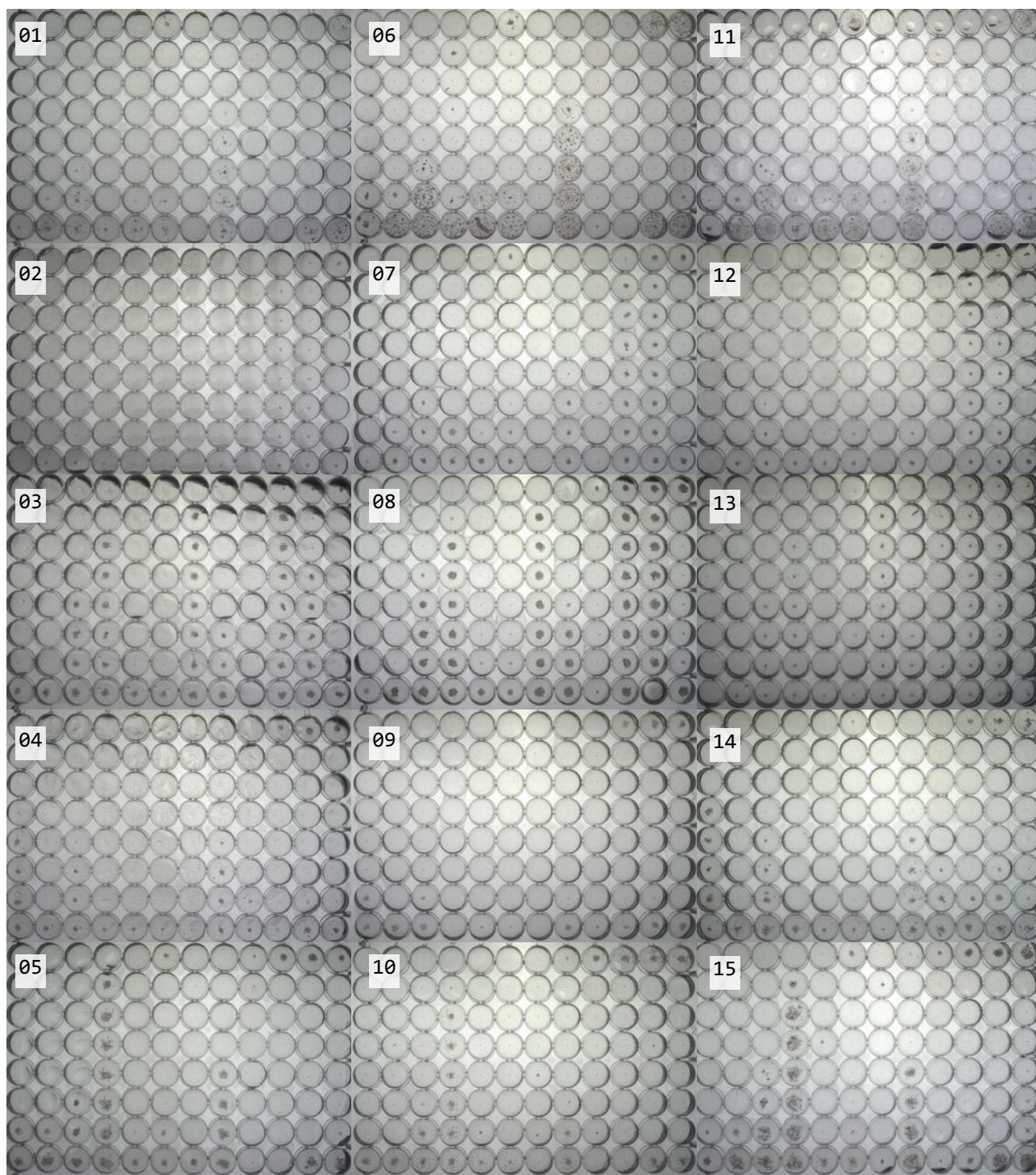


Figure S1: Related to Fig. 2. Raw images of fifteen plates taken from the testset. Each column corresponds to one of three laboratories and each row is a unique strain. All plates were inoculated and then incubated for 14 days according to Thermo Fisher's standard protocol as described in the Methods. The raw images captured by the Vizion™ Digital MIC instrument are greyscale, low resolution, noisy, unevenly illuminated and have poor contrast. Several of the test plates have artefacts, including shadows, sediment and air bubbles. All fifteen numbered images are included in the examples/ folder of the AMyGDA package so this figure can be reproduced.

Supplemental Methods

Software

The algorithm is written entirely in object-oriented Python. Each image is stored as a numpy array [1] and all image processing is done using the OpenCV2 Python API [2]. To allow for straightforward metadata storage we use the `datreant` module [3]. Specifically a new class (`PlateMeasurement`) is built off the standard `Treant` class provided by the `datreant` module; this stores all the metadata associated with each image in a JavaScript Object Notation (JSON) file. Crucially, the `datreant` module has a `discover` function that finds and reads all JSON files in a filesystem tree. This is much more flexible than using a database. One consequence, however, is the approach works best if each image is stored within its own subfolder. The software, with its accompanying academic use licence, a tutorial and all fifteen images in Fig. S1, can be downloaded from <http://fowlerlab.org/software/amygda>.

Assumptions

We assume that each well is inoculated in the centre and hence the bacteria will grow out radially from the centre. Hence, if any colonies form, they can be found by looking in the central region of each well. The software assumes the images are 8-bit greyscale, and hence the intensity of each pixel is simply characterised by a number in the range 0 to 255. We further assume that, in the centre of each well at least, dark pixels (i.e. pixels with low numerical values) represent bacterial growth and conversely light pixels represent no growth. We shall derive a numerical threshold for growth shortly.

Sources of error

False positive classification of growth is primarily the detection of *artefacts* in the image, whilst false negatives tend to be the algorithm missing small and/or faint patches of growth. These sources of error cannot be eliminated, merely minimised, since they are inversely coupled: using a less strict threshold to classify bacterial growth to reduce the number of false negatives will inevitably incorrectly classify more artefacts as growth, increasing the false positive rate. Let us now describe the types of artefacts that can be encountered in 96 well plate images.

Artefacts

Artefacts can be divided into two groups. The first systematically afflicts antibiotic lanes, or even a whole plate, whilst the second group tend to be stochastically distributed across a plate. Since the latter typically result in nonsensical growth patterns – the bacteria grows at high, but not low, concentrations of antibiotic – they can usually be detected by the software and the affected plate flagged. The first group are therefore more troublesome and we shall describe later how the growth detection logic attempts to minimise their effect.

The main systematic artefacts are shadows (Fig. 3a) and sediment (Fig. 3b). Shadows occur on the wells around the circumference of the plate because the ThermoFisher Vizion™ Digital MIC instrument uses a point source of illumination that is above the centre of each plate. A significant proportion of each shadow is usually as dark, or darker, than any bacterial growth, and therefore these artefacts cannot be distinguished based on hue alone. One could use the prior knowledge that shadows only affect the perimeter of each plate to identify when a plate is badly affected, unfortunately, the layout of this plate (Fig. 1) is such that *all* the wells for bedaquiline, kanamycin, ethionamide and PAS are at the edges of the plate and therefore, when there are particularly strong shadows, it is possible the algorithm could systematically and erroneously measure ‘growth’ in all these wells, thereby returning high MICs for these anti-TB drugs. In any future plate re-design the impact of shadows could

be mitigated by ensuring, where possible, the wells at the edges of the plate have low concentrations of antibiotic below the breakpoint. Then if AMyGDA incorrectly classified growth in these wells due to shadows there would be little or no impact on the diagnostic result.

A small number of images of plates have small dots in the centre of the majority of wells which do not diminish at high antibiotic concentrations and appear qualitatively different to any *M. tuberculosis* growth present (Fig. 3b). These are the remnants of the original inoculation and hence we call this artefact type ‘sediment’. Since, as shown here, an entire antibiotic lane is usually affected, then it is possible the algorithm will incorrectly measure growth in all wells and predict a high (and incorrect) MIC. Unlike shadows, this artefact can affect all the drugs on a plate and therefore it is very important to mitigate against it.

Air bubbles (Fig. 3c), condensation (Fig. 3d), contamination (Fig. 3e) and plate failure (Fig. 3f) are the main artefacts that stochastically affect wells. Air bubbles and condensation are similar; both form during inoculation of the plate and sometimes diminish or disappear during incubation. Even after two weeks, however, some remain. Air bubbles are small and highly localised (Fig. 3c) and hence are only a problem if they lie within the central measured region. Due to total internal reflection, the air bubbles usually appear as dark rings with a lighter centre and so can often be identified by a human as not being bacterial growth. The algorithm, however, cannot tell the difference and, if a well contains a sufficient number of dark pixels due to one or more air bubbles, the well will be incorrectly classified as containing bacterial growth. We also often observe features that resemble paint bubbles (Fig. 3d). These we call ‘condensation’. Usually the majority of pixels of these features are not dark enough to be classified as growth, however, occasionally enough are, leading to false positive readings. Both air bubbles and condensation can affect large areas of a single plate leading to a large number of false positives on a single plate.

A well can occasionally become contaminated by another microorganism which usually grows rapidly, leading to a well with qualitatively different growth (Fig. 3e). Such wells are very obvious to the human reader who may, depending on whether they affect the MIC, choose to ignore them. The algorithm, however, blithely classifies that well as containing a large amount of bacterial growth. Contaminated wells are rare and so usually lead to an nonsensical growth pattern (although this is not true for the example shown here). Finally, and very rarely, we see what appears to be nonsensical *M. tuberculosis* growth in wells with very high antibiotic concentrations (Fig. 3f). This, we assume, is due to that well not containing the specified antibiotic at the required concentration due perhaps to a quality control issue during manufacture, or degradation of the antibiotic on the well prior to use.

Filtering

Although all the raw images have been taken in the controlled environment of the interior of a ThermoFisher Vizion™ Digital MIC device, the images are noisy, there is notable variation in brightness between images and contrast is generally poor (Fig. S1). The latter can be demonstrated by examining the pixel intensity histogram of a whole image (Fig. 2a).

Three filters are used to help correct these deficits. First, a mean shift filter [4] is applied. This examines the pixels within a circular window (radius 10 pixels) and clusters them in colorspace. The procedure is iteratively applied to the entire image, resulting in a ‘flattening’, thereby removing much of the background noise (Fig. 2b & S2) but, crucially, not altering the sharpness of any edges, such as the circumference of the wells. The resulting image, however, still has poor contrast. Naively applying a global histogram equalization filter does indeed lead to a near-ideal flat distribution of pixel intensities, however, the filter has greatly accentuated the subtle differences in illumination across the plate. Instead, let us apply a form of local histogram equalization, called contrast limited adaptive histogram equalization (CLAHE, Fig. 2c & S3). This is similar to the global method, but divides the image

into tiles, which we here dynamically assign to have approximately the same size as the wells. The pixel intensities within each tile are then adjusted and finally the borders between tiles are interpolated to remove any tessellation artefacts. Compared to the original image most (but not all) of the variation in illumination has been removed and the contrast, as indicated by the broader histogram, is greatly improved with the result that the bacterial growth is much easier to see. Both of these filters are implemented using the OpenCV Python API [2]. The final filter attempts to ensure that all images have a similar brightness by shifting the stretching the pixel histogram such that the mode lies at a pixel value of 180 (out of 255 since it is an 8-bit grayscale image) and the 5th and 95th percentiles lie have values of 70 and 220, respectively. Usually this has little effect on an image (Fig. 2d, S4), but if, for some reason, the image is unusually light or dark, this manipulation can at least partly ‘rescue’ an image.

References

1. **Van Der Walt S, Colbert SC, Varoquaux G.** 2011. The NumPy array: A structure for efficient numerical computation. *Comp Sci Eng* **13**:22–30. <http://dx.doi.org/10.1109/MCSE.2011.37>
2. **Bradski G, Kaehler A.** 2008. *Learning OpenCV*. O’Reilly Media, Inc, first edition.
3. **Dotson DL, Seyler SL, Linke M, Gowers RJ, Beckstein O.** 2016. datreant: persistent, Pythonic trees for heterogeneous data. In *Proc 15th Python Sci Conf*, edited by S Benthall, S Rostrup, 51–56.
4. **Comaniciu D, Meer P.** 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Trans Pattern Analysis Mach Intel* **24**:603–619. <http://dx.doi.org/10.1109/34.1000236>

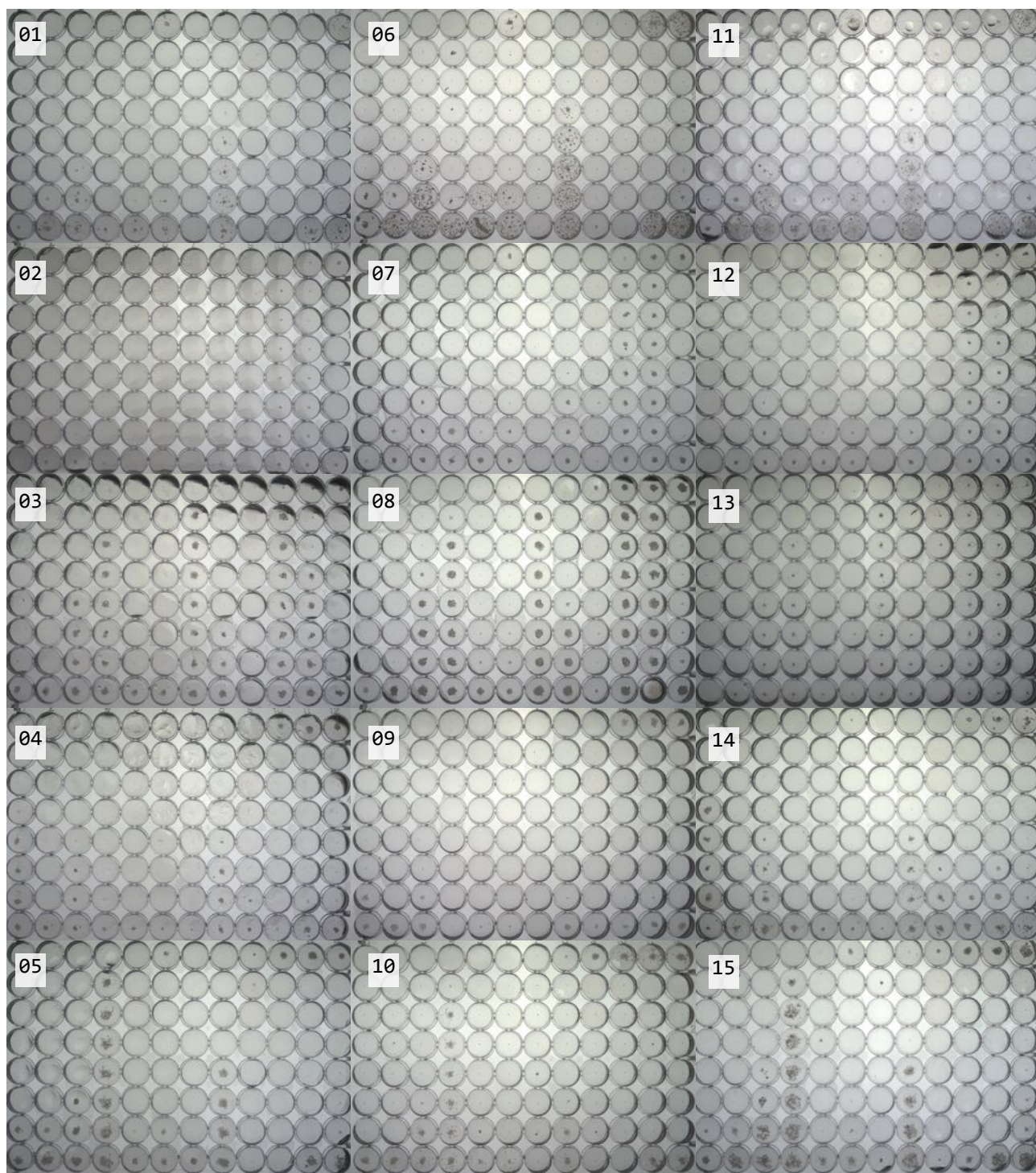


Figure S2: Related to Fig. 2. To reduce the background noise without affecting the sharpness of the edges, a mean shift filter [4], as implemented in OpenCV [2], is applied to all fifteen images in the testset. The images still have poor contrast and illumination. Each column corresponds to one of three laboratories and each row is a unique strain. All fifteen numbered images are included in the examples/ folder of the AMyGDA package so this figure can be reproduced.

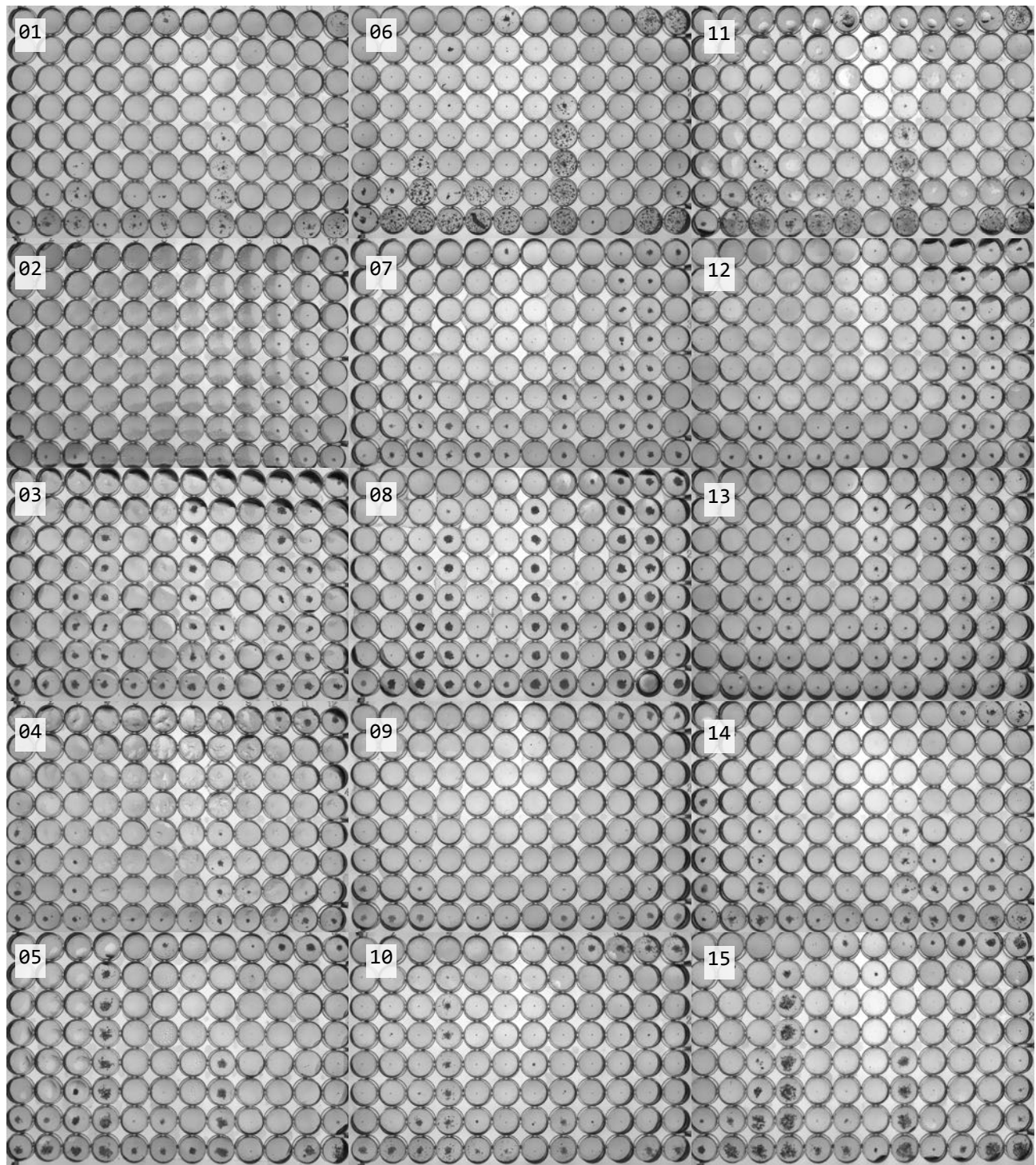


Figure S3: Related to Fig. 2. To both normalise the illumination and improve contrast, a contrast limited adaptive histogram equalization (CLAHE) filter is then applied, as implemented in OpenCV [2]. Contrast is greatly improved, making the bacterial growth easier to see, although there remains some uneven illumination across the plate. The tile size in the CLAHE algorithm is chosen to approximate the size of each well. Each column corresponds to one of three laboratories and each row is a unique strain. All fifteen numbered images are included in the examples/ folder of the AMyGDA package so this figure can be reproduced.

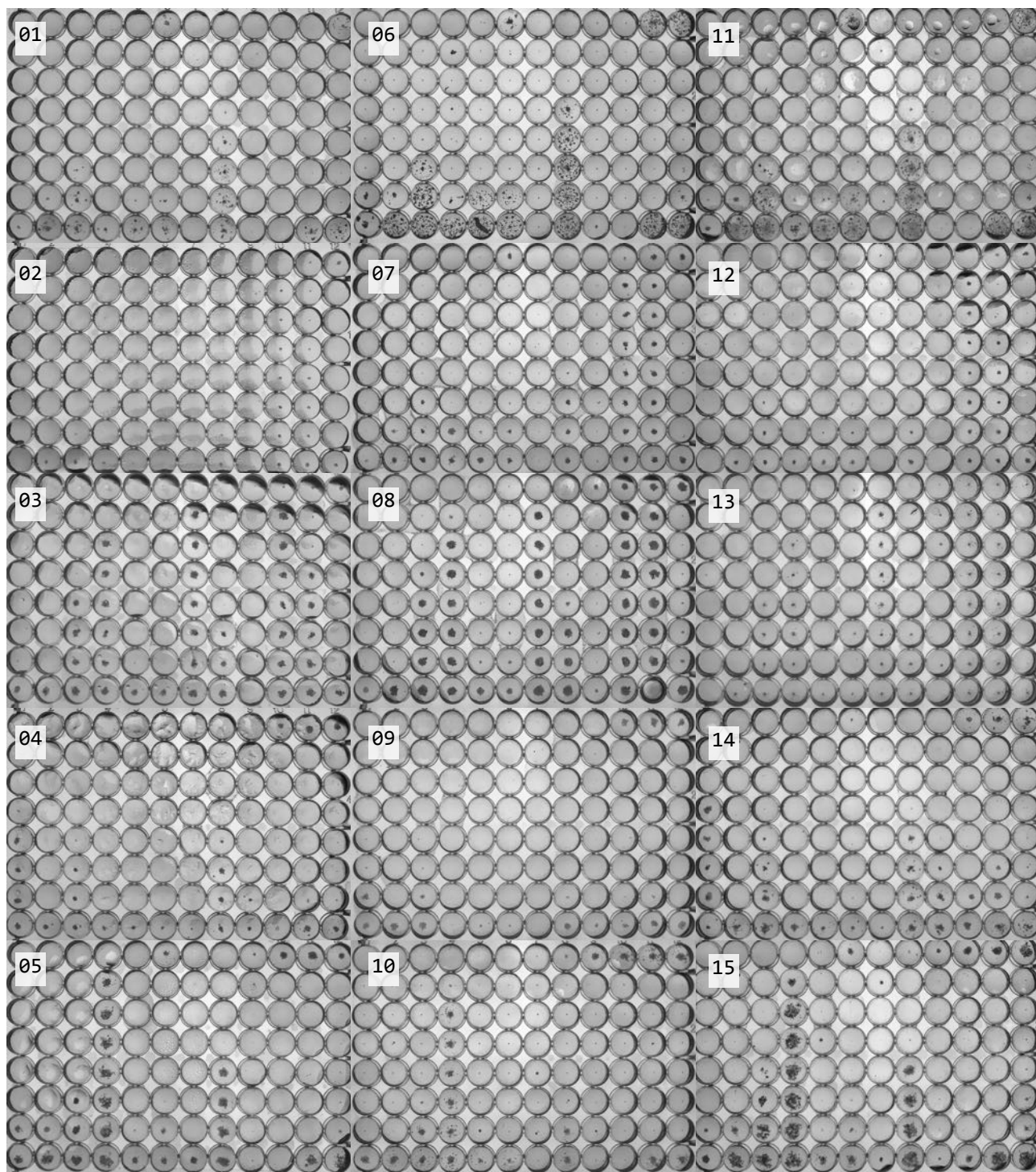


Figure S4: Related to Fig. 2. To normalise the brightness between images, the pixel histogram is shifted and stretched such that the mode is 180/255 and the 5th and 95th percentiles fall at pixel values of 70/255 and 220/255. If the Vizion instrument is setup as described, the effect is often minor, for example by comparing to Fig. S3 one can see slight changes especially in images 02, 11, 13. This step is especially important if an unusually light or dark image is captured which would otherwise lead to too little or too much growth being detected. Each column corresponds to one of three laboratories and each row is a unique strain. All fifteen numbered images are included in the examples/ folder of the AMyGDA package so this figure can be reproduced.

Identifying the wells

Despite the 96 well plate being held in a drawer within the Vizion™ Digital MIC instrument, the orientation and position of the plate changes subtly from image to image and therefore it is not possible to assume each well will always be in the same location on the image. The next task is therefore to identify the locations of all 96 wells on each plate. This is done using a Hough transform optimised for circles as implemented in OpenCV [2]. We have some prior information: each plate contains exactly 96 circles, one for each well, and we also can estimate the approximate radius and location of each well from the dimensions of the image. The algorithm first detects all circles in the image with a radius within a specified but narrow range. The number of circles are counted, and if fewer than 96, the range is broadened slightly and the process repeated. When exactly 96 circles are found, the code then checks to see if there is one, and only one, circle within the expected region of each well. Only then does the algorithm report success. Due to these stringent criteria this iterative approach works well (Fig. S11a) and only failed in tests when the image was so badly aligned that fewer than 96 wells were visible (Fig. S5).

The approach does, however, suffer from one problem that will become important later when we discuss deriving the bacterial growth detection logic. At the edges of the plate, away from the optical axis of the camera, the images of the wells have *two* overlapping circles corresponding to the top and bottom of the well. The position of the centre of the well is not well-defined in these cases, since we are implicitly assuming throughout that we have a perfect two-dimensional image of each plate. Our approach will stochastically select one of circles, thereby introducing some uncertainty into the position of the well centre, which increases we move further from the centre of the plate.

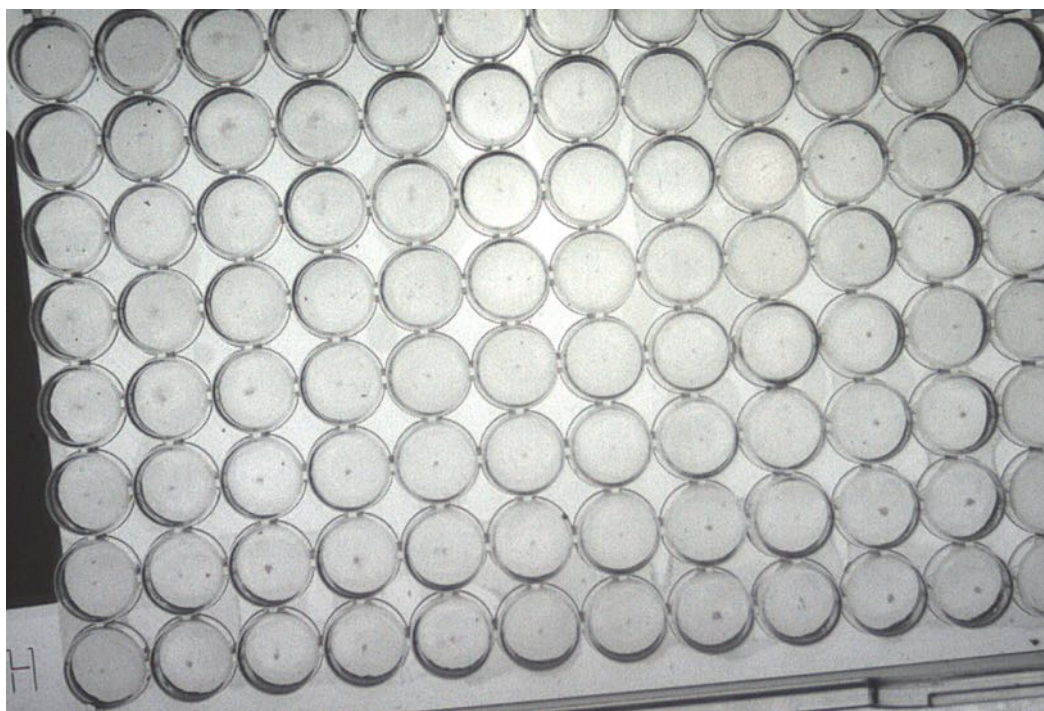


Figure S5: Related to the Methods. The algorithm failed to detect the wells in this image since there are fewer than 96 wells visible. This is a ‘good’ failure since the automated detection of growth should cease at this point and the plate be rejected.

The Bacterial Growth Detection Logic

The determination of a growth detection logic is a complex and iterative process since the choice of one parameter tends to affect the others. We shall therefore describe our final, optimised growth detection logic before justifying it. To minimise the incorrect detection of shadows, the algorithm only measures growth within a circle placed in the centre of the well whose radius is 50% of the estimated diameter of the well. A well is classified as containing bacterial growth if more than 2% (or a quarter of the average growth in the control wells, whichever is higher) of the pixels within this circle have an intensity less than 130 (out of 255 since these are 8-bit images).

Due to the possibility of shadows and the inherent uncertainty in where the edge of each well is, trying to measure the amount of growth in the ‘whole’ well would be very prone to false positives. It is reasonable, however, to only consider a central region of each well, since it has been inoculated in the centre and the growth of any *M. tuberculosis* should be approximately radial. Choosing the size of this central region is a tradeoff: too large and it may include shadows. Too small and some real bacterial growth may be missed. These effects can be seen in Fig. S6 where the effect of varying the radius of the circle from 40% to 60% of the well diameter on five plates chosen from the test-set, all of which are affected by shadows. The middle value of 50% is not perfect – a few shadows are still incorrectly classified as growth – but appears to be a reasonable compromise between the two competing effects.

A well is defined as containing bacterial growth if the proportion of pixels classified as being growth is greater than a specified threshold. This corresponds to choosing a pair of parameters: the pixel intensity below which a pixel is counted as being bacterial growth (the pixel intensity threshold) and the minimum proportion of such pixels required for a well to be classified as containing bacterial growth (the growth threshold). We first divided all wells into two sets based on the MICs measured by the experienced lab technician using the Vizion™ Digital MIC instrument. For example, if for one plate the MIC recorded for isoniazid corresponded to the fourth well in the seven series wells on a plate, then the first three wells should all contain bacterial growth and were added to the growth set. The remaining four wells should contain no bacterial growth and were added to the set of no growth. The cumulative pixel intensity histograms were then calculated for all wells in each set and plotted (Fig. S7a, b & c). As expected, the cumulative histograms for the wells containing growth have, on average, more dark pixels (Fig. S7d).

We then systematically varied the pixel intensity and growth thresholds and calculated the true and false positive and negative rates, storing them as two-dimensional arrays (Fig. S8a). As expected, it is trivial to have very relaxed thresholds and identify all wells with growth, but at the cost of a very high false positive rate. Conversely, very strict thresholds eliminate the false positives, but at the cost of very high false negatives: optimal values of the parameters will lie somewhere in between these two extremes. Arrays of the sensitivity and specificity vary can be straightforwardly calculated (Fig. S8b). We chose to simply optimise the product of sensitivity and specificity (Fig. S9), thereby identifying a small region (Fig. S9b) in our parameter space, which our chosen parameter pair (130, 2%) lies at the centre of. Note that we are not suggesting that our chosen parameters are *the* optimal pair, merely that they belong to a set of parameters that maximises the product of sensitivity and specificity. For our test-set, we recover a sensitivity of 86.1% and a specificity of 87.2%.

Testing revealed that this logic, however, was still prone to incorrectly classifying sediment and air-bubbles as growth (Fig. S10). Since these artefacts are typically small, we cannot easily distinguish them from bacterial growth when the latter is also small. When there is good bacterial growth, however, one could, in principle, use the difference in the amount of growth to isolate the artefacts. We therefore tested making the growth threshold

a function of the average amount of growth in the control wells. Since one would expect the growth in a well containing antibiotic to be a less than the control well growth, we tested making the growth threshold a range of fractions of the growth in the control well, from a half down to a fifth (Fig. S10). As one would expect, this is a tradeoff between not detecting the artefacts but also not missing real bacterial growth. Our simple study suggested that a dynamic growth threshold of 25% of the average growth in the control wells was a good compromise. To prevent the growth threshold being set too low, this logic only applies if the resulting growth threshold is $> 2\%$.

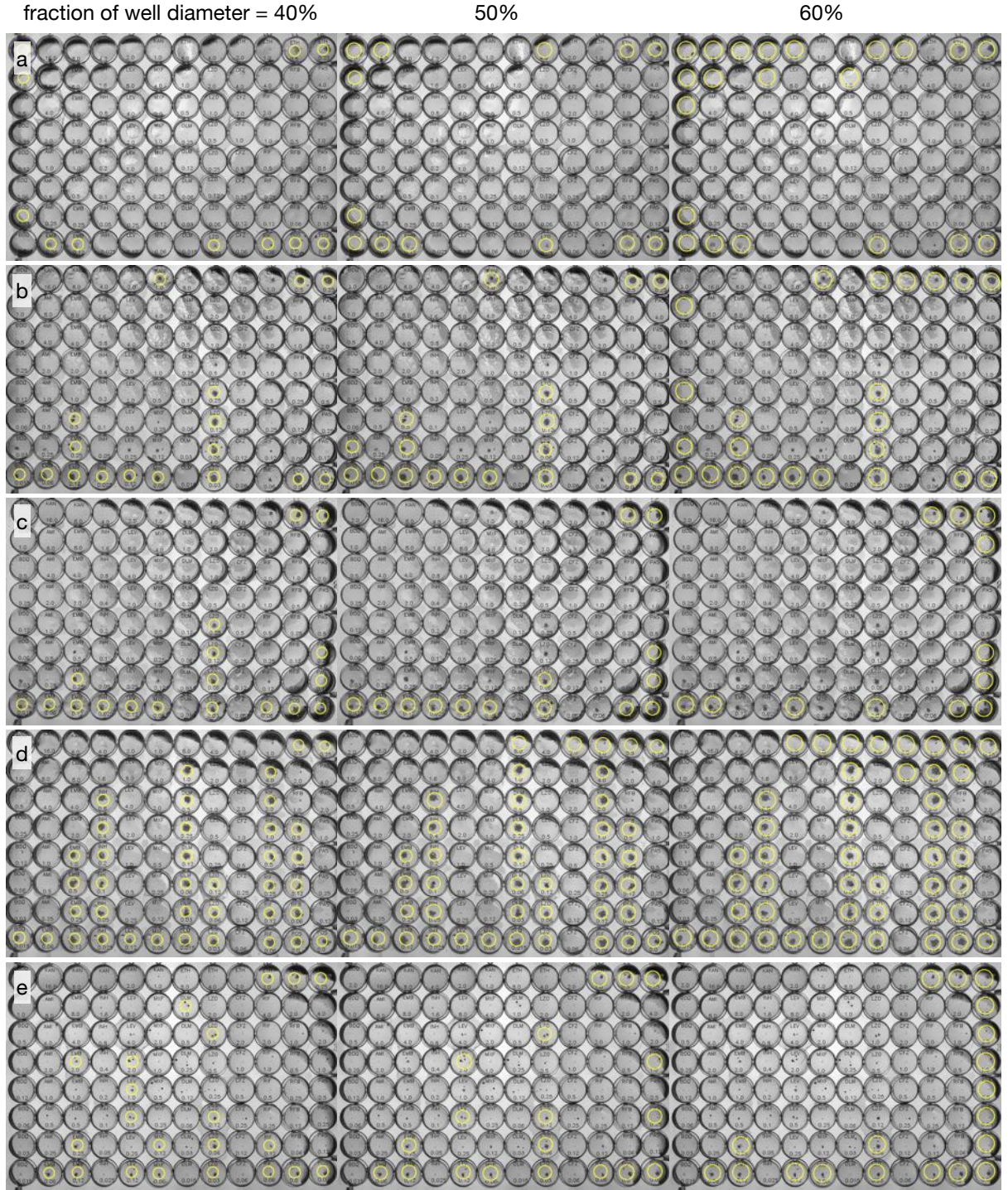


Figure S6: Related to the Methods. Only the centre of each well is assessed, as defined by a circle whose radius is a fraction of the measured well radius. There is a trade-off: small circles avoid shadows, and so minimise the effect of this artefact, but large circles have a better chance of capturing any bacterial growth that is away from the centre of the well. Shown here are five images from the testset which are afflicted by shadows to varying degrees. The effects of using three different radii, 40%, 50% and 60%, are shown.

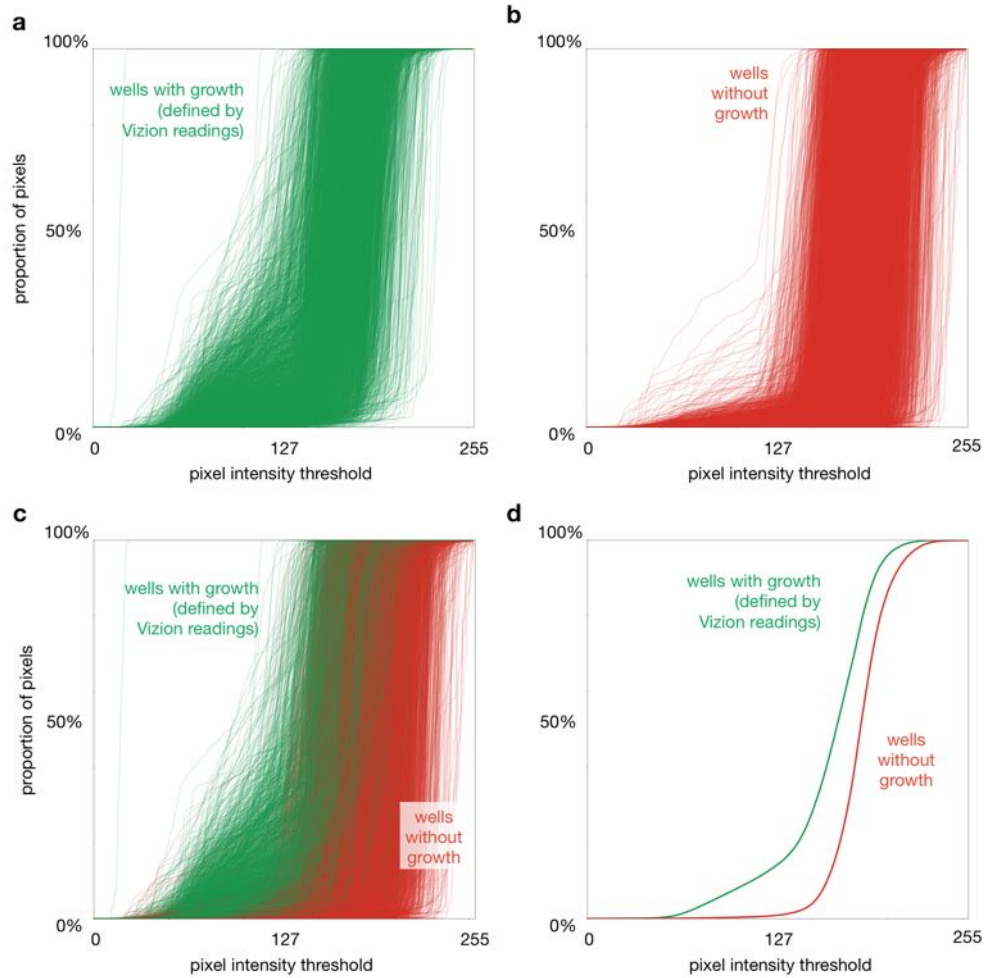


Figure S7: Related to the Methods. The wells characterised as containing growth by the technicians are different, as measured by the cumulative pixel intensity histogram. Each well can be characterised by the cumulative histogram of the pixel intensities. The histograms for (a) all the wells that according to the MIC measurements performed by the technician contained growth and (b) all the wells that do not. (c) For comparison, all the histograms are plotted on the same graph. (d) The average cumulative histogram pixel intensity histograms. Intensities run from 0 (dark) to 255 (light).

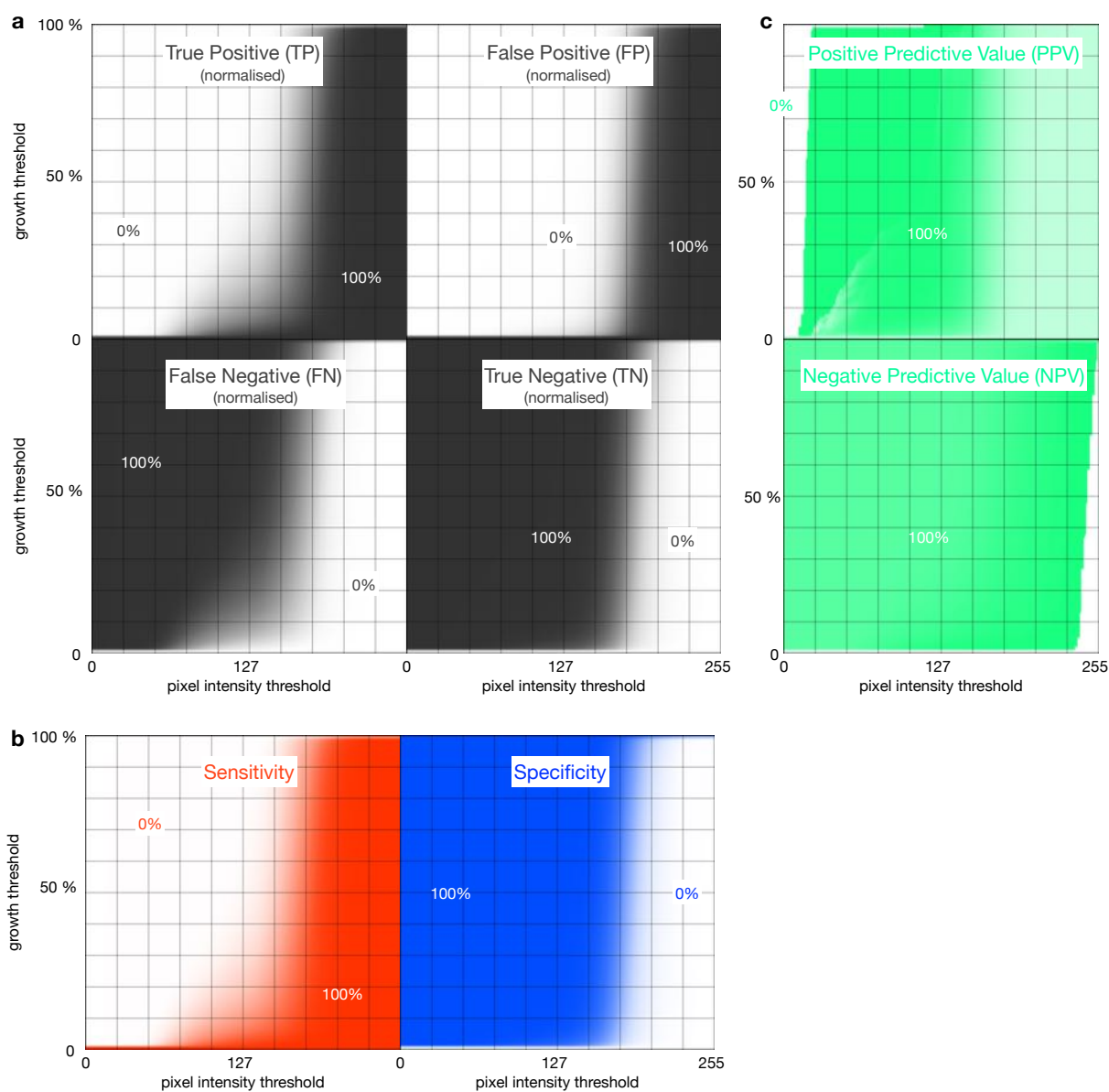


Figure S8: Related to the Methods. All the histograms in Fig. S7 can be converted into arrays. **(a)** The four arrays characterising the classification, true positive (TP), true negative (TN), false positive (FP) and false negative (FN), are plotted. All are normalised. **(b)** The sensitivity and specificity. **(c)** The positive and negative predictive values (PPV and NPV).

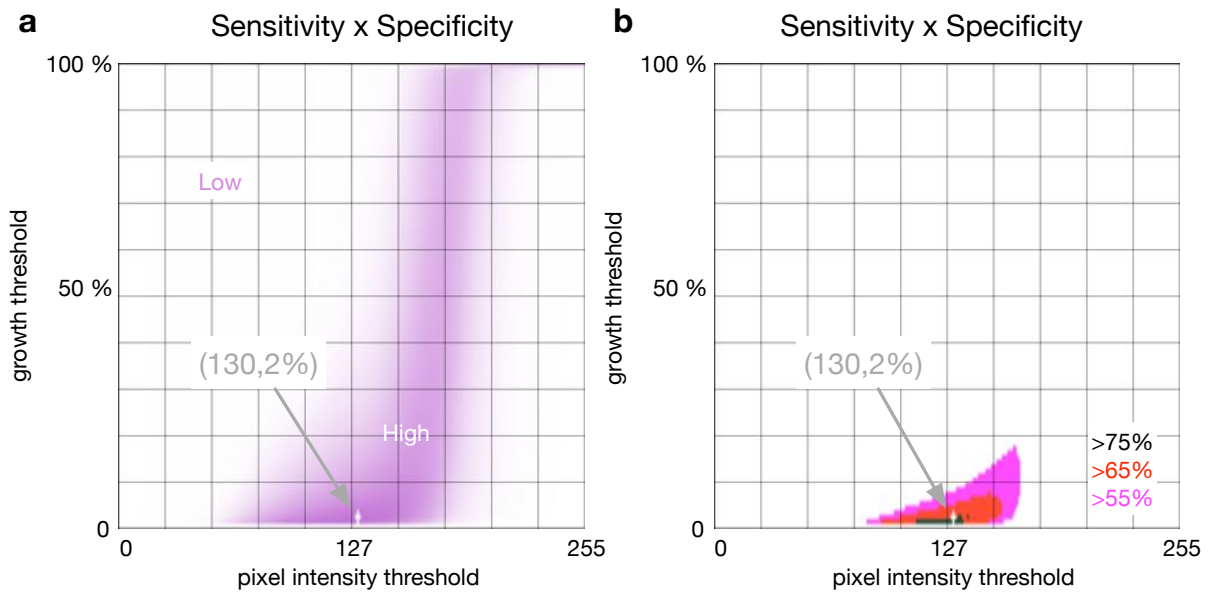


Figure S9: Related to the Methods. The parameters defining growth in a well were chosen by maximising the product of sensitivity and specificity. Any pixel with an intensity less than 130 (out of 255) are classified as bacterial growth. If a well contains more than 2% such pixels it is classified as containing growth. **(a)** The product of the sensitivity and the specificity with the final chosen parameters labelled. **(b)** The final chosen parameters are in the centre of the maximum of this product. Shown are the regions where the product is greater than 55%, 65% and 75% are shown. For the testset the chosen parameters have a sensitivity of 86.1% and a specificity of 87.2%.

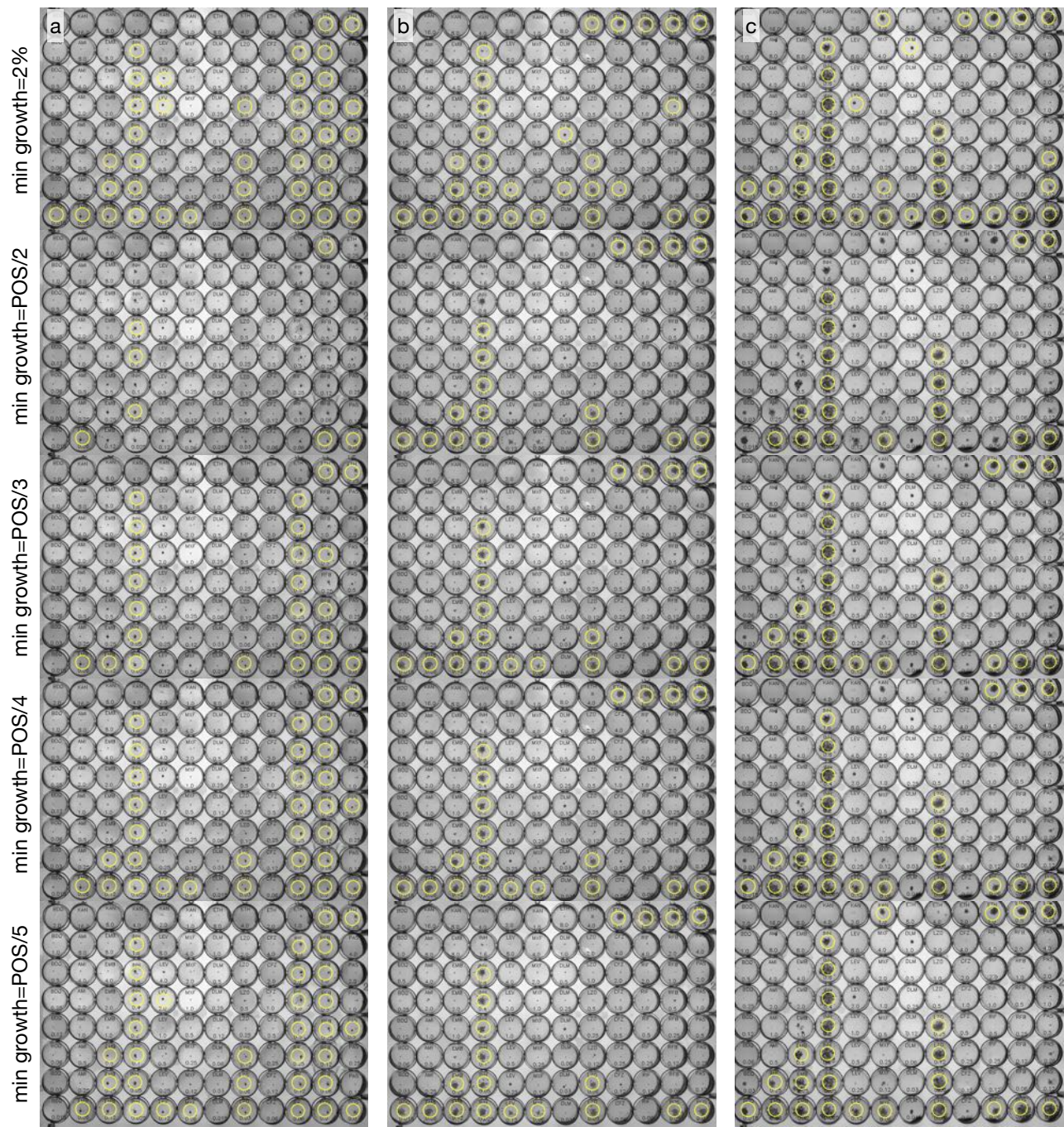


Figure S10: Related to the Methods. Making the definition of growth a function of the amount of growth in the control wells helps minimise the incorrect detection of artefacts. Three images from the testset, each afflicted by sediment and air bubbles are shown. In the first row any well where more than 2% of the pixels in the central circular region are below the threshold are classified as containing growth. This incorrectly classifies several of the wells with the artefacts as containing growth. The average growth in the positive control wells is first measured. The growth threshold can then be defined as a fraction of this growth, with 2% as a lower limit. Shown are a range of fractions of the growth in the control wells. As the fraction is decreased, the number of correct classifications increases, until a point is reached at which artefacts begin to also be classified.

The final composite image

Once the coordinates of all 96 wells are known, it is straightforward to annotate the image with additional useful information. Since the plate design of UKMYC5 is slightly complex (compared to e.g. MYCOTB), it is helpful to mark the location of each of the antibiotic lanes (Fig. S11b). Likewise, labelling each well with the abbreviated name of each antibiotic, and its concentration (Fig. S11c), ensures the plate design is recorded in the same place as the observed bacterial growth. Finally, the wells where growth of *M. tuberculosis* is detected are labelled with a coloured circle – this also defines the region that was analysed for growth. All this annotation is then composited with the final filtered image, as shown Fig. S11e, which if stored appropriately, provides a complete audit record of the process.

Performance

The AMyGDA code was benchmarked on four different computers, three times each. In decreasing order of CPU speed these were a workstation (with a 2013 Intel Xeon Ivy Bridge CPU), a desktop, a laptop and a Raspberry Pi 2. The mean time to analyse a single image on each machine took 3.7, 5.0, 5.5 and 40 seconds, respectively. On all architectures, the three most computationally expensive tasks were, on average: (i) applying the mean shift filter (62%), (ii) starting the python interpreter and loading the modules (17%) and (iii) identifying the location of all 96 wells using the Hough transform (17%). With the exception of the Raspberry Pi 2, the code is therefore fast enough for interactive use, even on inexpensive, commodity hardware. In practice, even the performance of the Raspberry Pi 2, which only costs \$40, is satisfactory as it analyses an image in less than the time an operator needs to replace the 96-well plate, in this case within a Vizion™ Digital MIC viewing instrument.

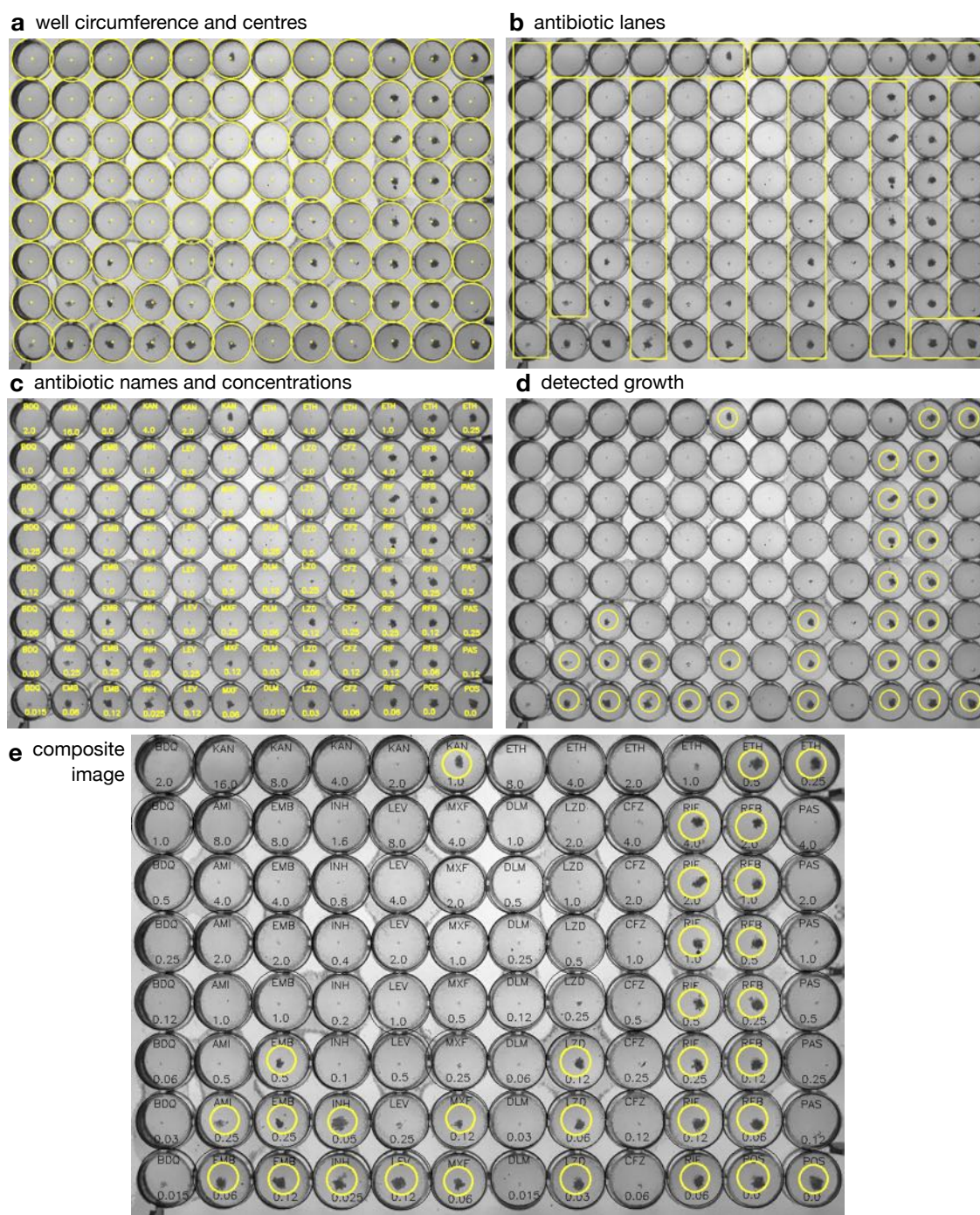


Figure S11: Related to Fig. 2. The software produces an annotated composite image with all detected growth labelled. **(a)** The circumferences of all 96 wells (and hence the centres) are first detected using a Hough transform optimised for circles. **(b)** It is then straightforward to either annotate where all the antibiotic lanes are, as shown here, or produce images of each antibiotic lane cut from the larger image, as shown in Fig. 3. **(c)** Likewise each well can be labelled with the concentration of antibiotic it contains to help with interpretation. **(d)** The wells which the algorithm classifies as containing growth are then marked by a coloured circle: this circle also defines the region of each well that was examined for growth. **(e)** Finally, all these features can be composited onto a single, filtered image that provides a complete record of the process. For clarity only the growth is drawn in yellow, all other features are in black.

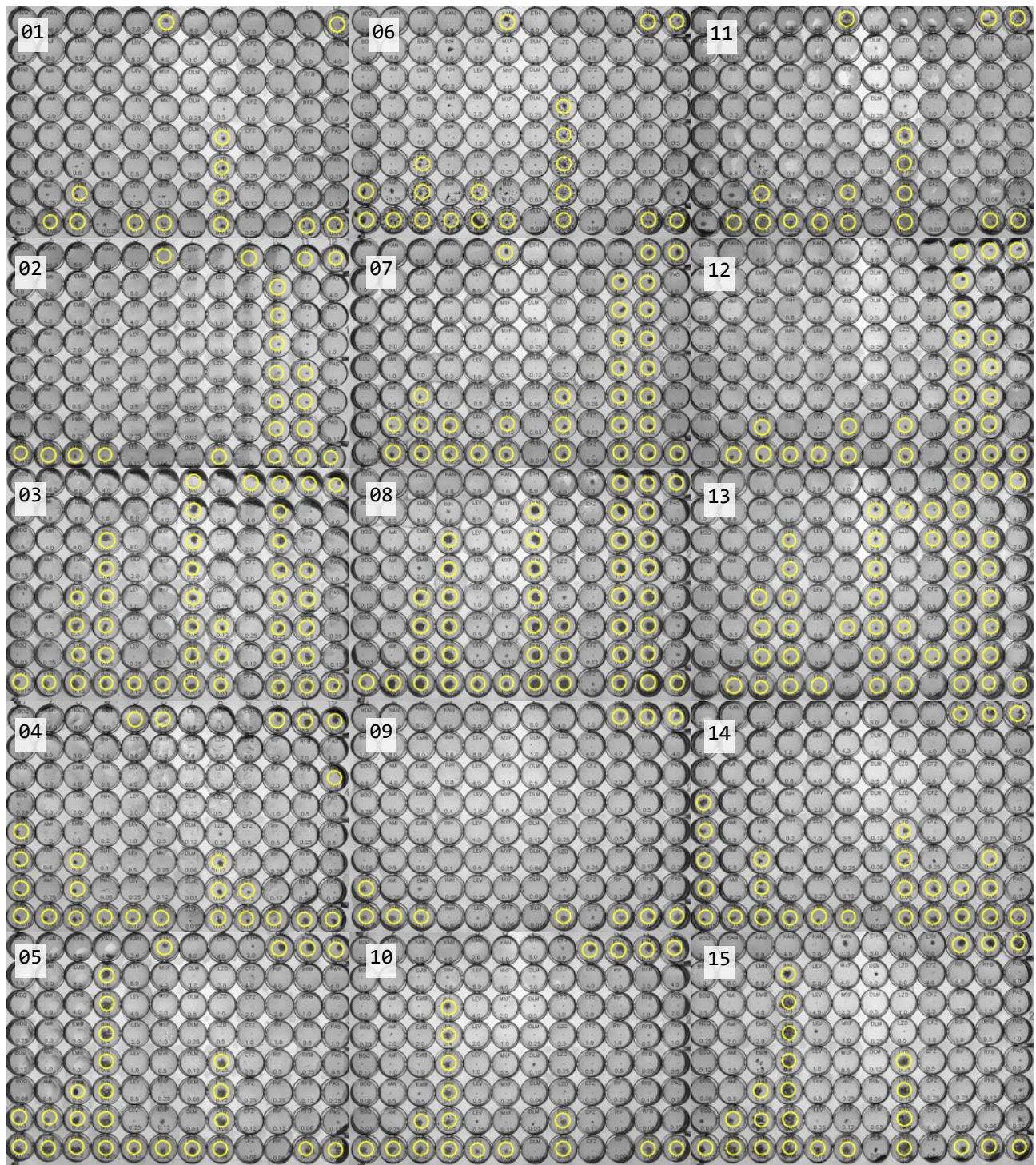


Figure S12: Related to Fig. 2. Following application of the filters, the circumference of each well is detected using a Hough transform that is optimised for circles, as implemented in OpenCV [2]. The bacterial growth is measured in each well, and if above a threshold, a yellow circle is drawn, as here. The circle also defines the region of each plate within which growth is detected. Each column corresponds to one of three laboratories and each row is a unique strain. All fifteen numbered images are included in the examples/ folder of the AMyGDA package so this figure can be reproduced.

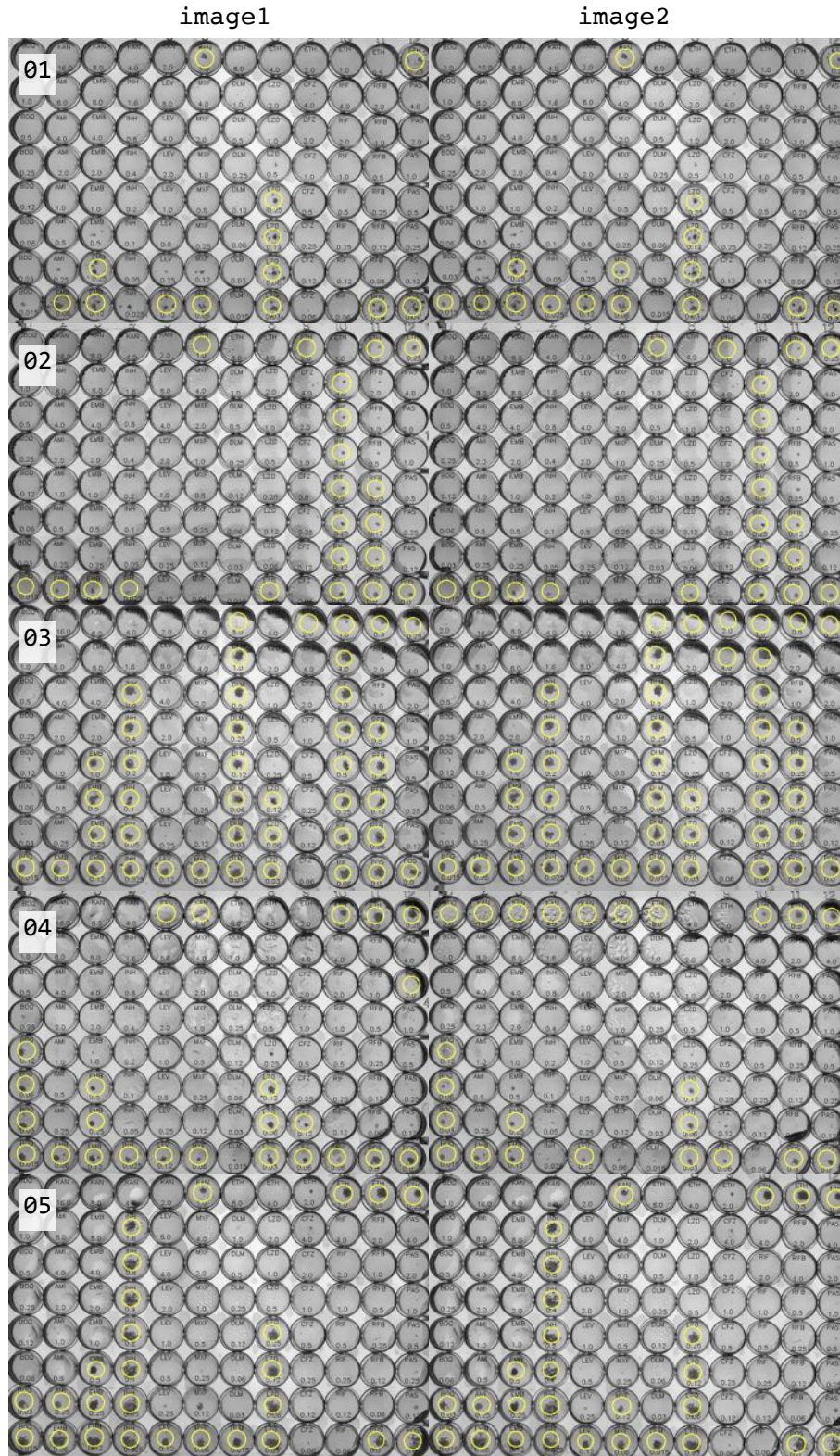


Figure S13: Related to Fig. 4. To test the reproducibility of the algorithm, all the plates in one of the three laboratories had a second photograph (image2) taken by a second technician using the same Thermo Fisher Vizion™ Digital MIC instrument. Hence in our example set of fifteen images, the first five have a second photograph that we can analyse using AMyGDA. These are shown and small differences in detected growth can be seen. In particular the condensation at the top of the plate in 04 is misclassified in the image2.