

Contour Area Filtering of Restriction Landmark Genomic Scanning 2-Dimensional Electrophoresis Profiles

¹Ramakrishnan Kazhiyur-Mannar, ³Dominic J Smiraglia,

²Christoph Plass, ^{1,4}Rephael Wenger

¹Computer Information Sciences Department, The Ohio State University, Columbus, Ohio, 43210

²Division of Human Cancer Genetics, Department of Molecular Virology, Immunology and Medical Genetics, and Comprehensive Cancer Center, The Ohio State University, Columbus, Ohio, 43210;

³Department of Cancer Genetics, and Comprehensive Cancer Center, Roswell Park Cancer Institute, Buffalo, NY 14263

Running Title: Contour Area Filtering of RLGS Profiles

⁴**To whom correspondences should be addressed:**

Rephael Wenger, Dept. of Computer & Information Science, The Ohio State University, 2015 Neil Ave., Columbus, Ohio 43210

Abbreviations:

Keywords: RLGS, image processing, isocontour, watershed filter

Abstract

Restriction Landmark Genome Scanning (RLGS) is a 2-dimensional gel electrophoresis technique for detecting DNA molecular changes that occur within restriction fragments. A typical RLGS profile consists of an electrophoretogram or a phosphor image of approximately 2500 spots. The profiles are quantitative with spot intensities reflective of the copy number of the DNA fragment resulting in a variety of spot intensities, particularly when applied to cancer genomes. The background intensity can vary widely across the image caused both by variation in spot density and by the physical laboratory process of creating a gel. We describe an algorithm based on contour areas for distinguishing between background and foreground in an RLGS profile. We present experimental results which show that contour area filtering is a quick, efficient method for separating background from foreground with extremely high accuracy.

Introduction

Restricted Landmark Genome Scanning (RLGS) is a 2D gel electrophoresis technique developed by Hatada et. al.[1] for detecting DNA molecular changes that occur near restriction enzyme sites. Genomic DNA is digested by a “landmark” restriction enzyme (i.e., *NotI* or *AscI*) and radioactive nucleotides are incorporated into the cleavage sites. The fragments are further digested by a second enzyme (i.e., *EcoRV*) and separated along one dimension using agarose gel electrophoresis. A third enzyme (i.e., *HinfI*) is used to digest these fragments in gel followed by second dimension separation via polyacrylamide gel electrophoresis. Autoradiography or phosphor imaging is applied to the dried gel producing an “RLGS profile” of approximately 2500 spots (Figure 6.) DNA fragments correspond to spots on this profile.

RLGS is used to detect methylation changes caused by cancer (for review see [2].) Methylation sensitive landmark enzymes (i.e., *NotI* or *AscI*) cut the DNA only at unmethylated sites. RLGS profiles are created using DNA from tumorous and compared to RLGS profiles from normal tissue. Missing, added, or amplified spots indicate DNA methylation changes or DNA copynumber changes which may occur in the cancer and are potential biomarkers of the tumor.

Two-dimensional gel electrophoresis is a standard technique in protein analysis. Extensive research and software has been developed for automatic analysis of protein profiles [3-8]. Commercial software includes ImageMaster, Melanie, PDQuest, and

Phoretix among others [5, 6, 8, 9]. These packages provide excellent user interface, statistical and database tools for assisting in gel analysis. They have more difficulty in fully automating the detection and identification of spots on the gel profiles. Phoretix supports the automatic, simultaneous analysis of large number of similar gels, using statistical tools to correct for errors in spot detection and identification of any individual gel.

Analysis of RLGS gels from genomic DNA poses certain challenges compared with analysis of protein gels. Spots are often lighter and smaller than protein spots making identification of individual spots more difficult. It is often the lightest spots, their existence or lack thereof, which is of most interest in detecting DNA methylation. Moreover, samples of tumor tissue almost always contain some normal tissue, creating faint images of spots from the normal DNA which are affected, or methylated, in the tumor DNA. In addition, RLGS gels typically contain over 2500 spots (although this is enzyme dependent) which is the high range for gel analysis.

Two software packages have been developed specifically for the analysis of RLGS gels, RAT (RLGS Analysis Tool) by Sughara et. al. [10] and DNAInsight by Takahashi et. al. [11-14]. Neither package is in widespread use by laboratories doing RLGS analysis, although a commercial version of DNAInsight has been announced.

Numerous algorithms and techniques are used for filtering background pixels and identifying spots in protein and RLGS gels. Sternberg [15] gives a background subtraction algorithm for protein gels using 3D gray scale morphological operators. Takahashi et. al. [11, 13, 14] use these gray scale morphological operators for

background subtraction and then apply a “ring operator” to identify individual spots and their centers. They subtract these spots from the image and reapply their “ring operator” to further identify hidden spots. Sugahara et. al. [10] use local thresholds to remove background pixels from the image. Melanie [4] uses thresholding of the second derivative of the gray scale intensities to identify foreground pixels. ImageMaster compares pixel intensities to intensities on the boundary of a surrounding window to identify foreground pixels.

The filtering methods described above have two major drawbacks. First they all require the setting of some sensitivity threshold related to the gray scale intensity of the spots. Users must often adjust these parameters for individual gels. Second, because of this sensitivity thresholding, the algorithms may fail to detect the lightest spots. Many of the algorithms implicitly smooth gray scale intensities, causing these lightest spots to “wash out” with the background. In addition, full intensity saturated spots have to be handled specially by some of the algorithms.

In this paper we present an algorithm for separating foreground from background in RLGS profiles using isocontours of the gray scale image. We identify isocontours which enclose regions of a specified area and select those regions as foreground. We do not actually construct the isocontours, only the set of pixels contained by the isocontour. Standard pixel image processing techniques are also used to remove noise from the images.

Our isocontour based algorithm does not rely upon any gray scale sensitivity threshold. In fact, there is no sensitivity parameter as input to our algorithm. Instead, the primary input

to our algorithm is the maximum area of a cluster of overlapping spots. This metric depends upon spot density and is much more robust across gels than spot intensity. Our algorithm can detect even the faintest spots as long as they are not in regions of high spot density. Faint spots which are adjacent to large clusters of darker spots can be missed by our algorithm but such hidden spots pose a problem for almost all algorithms.

The next step after background subtraction is segmentation of the foreground into individual spots. Image segmentation is much more difficult than background subtraction. Contour area filtering can be used to perform some of this segmentation although with less impressive results, and other techniques are needed to complete this segmentation.

We applied our algorithm to three different types of RLGS gels and demonstrate that it performs exceptionally well, correctly identifying most of the background pixels and only rarely misidentifying spot pixels as background pixels. We compare results from our algorithm and from ImageMaster and demonstrate the superiority of our algorithm for filtering RLGS gels. Our isocontour filtering algorithm is one component in an automated RLGS gel analysis system under development.

Materials and Methods

RLGS gels

RLGS gels for both mouse and human genomic DNAs were run as previously described in [16], and modified as described in [17].

The autoradiograms are scanned at 300 dots per inch and stored as a tiff image of 5100 x 4200 pixels with 8 bits per pixel representing a gray scale in the range 0 to 255.

Contour Area Filtering Algorithm

The contour area filtering algorithm is shown in Figure 1. If a pixel p with intensity I_p is foreground then all the adjacent pixels with intensity greater than or equal to I_p should also be foreground. All their adjacent pixels with intensity greater than or equal to γ should also be foreground. Consider the maximal connected component containing p and pixels with intensity greater than or equal to γ . (Use 4-connectivity, connecting a pixel to the pixels to the left, right, above and below.) If p is foreground, then all the pixels in this component are also very likely foreground. If this component is very large, then it is a good indication that pixel p is not foreground.

If we replace the pixels by a continuous scalar field, then we can replace the maximal connected component by the area enclosed by an isocontour through p . An *isocontour* is a curve consisting of points with the same scalar value. If the area contained by the isocontour through p is large, then we mark p as background. This area can be thought of as the “watershed” of p (where intensity represents depth) and is used in many similar “watershed” based algorithms.

For each pixel p with intensity γ , let A_p be the number of pixels in the maximal connected component containing p and pixels with intensity greater than or equal to I_p . We mark all pixels with A_p greater than some user specified threshold as background.

Of course, if we compute A_p for each pixel separately, the algorithm would be far too slow. Instead we compute A_p in a single pass by slowly growing components starting at their most intense pixels. For each pixel, create a set containing only that pixel. Sort the pixels by intensity in decreasing order. Sorting the pixels by intensity also sorts the set of pixel intensities in decreasing order. For each intensity γ in decreasing order, make two passes over the set of pixels with intensity γ . First, for each pixel p with intensity γ , union the set containing p and sets containing pixels adjacent to p (left, right, top, bottom) with intensities greater than or equal to γ . This forms maximal connected components of pixels with intensity at least γ . Next, for each pixel p with intensity γ , store the size of the set containing p . This size is A_p .

We use a slight modification of the standard union-find data structure to represent the sets of pixels. The data structure is represented in an array, U , of pointers, one for each pixel. A set is represented by a tree of pointers, pointing back to the root. $\text{FindSet}(q)$ returns the element at the root of the tree containing q by following pointers $U[q]$ back to the root. It also performs “path compression” by resetting the pointers along the way to point to the root. To form the union of two trees rooted at p and s , we simply set $U[s]$ equal to p . Because of the order in which pixels are processed, pixel p in statements 10 and 11 is the root of the tree containing p and so $A[p]$ is the size of the set containing p .

The algorithm requires the input array, I , of pixel intensities and two other arrays U and A , containing the pointers and the contour area sizes. It runs in worst case $O(n \log(n))$ time where n is the number of pixels. A modification which requires one more field per pixel can guarantee $O(n \alpha(n))$ running time where $\alpha(n)$ is the inverse of Ackermann’s

function. In practice, the algorithm seems to take linear time and the modification is unnecessary.

The running time and space for sorting pixels depends upon the type of sort used. Our images are 8-bit gray scale consisting of only 256 pixel intensities and so bucket sorting will sort the pixels in $O(n)$ time using one array of size n . For larger sets of intensities, a more general $O(n \log n)$ sorting algorithm can be used.

Contour_Area_Filter is a very conservative procedure which cannot distinguish between noise and foreground. After applying Contour_Area_Filter, we use some standard morphological operators to remove some of the noise from the foreground. First, we apply the opening operator (erosion followed by dilation) to remove tenuous connections between pixels. Second, we remove any remaining “salt and pepper” noise by identifying very small foreground connected components and marking them as noise.

Segmentation

Contour areas can also be used to segment foreground into individual spots. The general idea is that each RLGS spot has a “center”, usually consisting of a relatively small number of high intensity pixels. If the area of a contour is approximately this size, then this contour surrounds a center.

Some spots, particularly saturated ones consisting of maximum intensity pixels, contain a large set of pixels with maximum or near maximum intensity. Thus if all the pixels in a contour have approximately the same intensity, then we also identify that contour as containing a center, even though the contour area may be very large.

Our algorithm for identifying center pixels is quite similar to Contour_Area_Filter. For each pixel p , we calculate and store the area $A[p]$ of the contour C passing through p . We also calculate and store in an array D the maximum intensity of any pixel in the contour passing through p . We initialize $D[p]$ to the intensity of pixel p . As we union p and a neighbor $D[q]$, we set $D[p]$ to be the maximum of its current value and $D[q]$. We again make a second pass over pixels with the same intensity, setting $D[p]$ equal to $D[r]$ where r is the root of the tree containing p .

We use a minimum center area and minimum center depth as two thresholds for identifying center pixels. Any pixel p with $A[p]$ less than the minimum center depth is marked as a center pixel. Any pixel p whose intensity differs from $D[p]$ by less than the minimum center depth is also marked as a center pixel. Each maximal connected component of center pixels forms a spot with that center.

Results

Contour area filter algorithm accuracy assessment

We ran our filter on master RLGS profiles created using enzyme combinations NotI-EcoRV-HinfI and AscI-EcoRV-HinfI on human DNA and NotI-EcoRV-HinfI on mouse DNA. (See Figure 6.) The human DNA is from the peripheral blood lymphocytes (PBLs) of a single healthy female donor. The mouse DNA is a combination of DNA from mouse strains FVB, C57/BL6J, and 129/SV. The master gels are used as a reference for all other gels with matching enzyme and genome in Dr. Plass's lab and have been extensively analyzed. We digitized autoradiograms of the gels at 300 dots per inch, creating tiff images of 5100 x 4200 pixels with 8 bits per pixel representing a gray scale

in the range 0 to 255. We implemented and tested `Contour_Area_Filter` on a 2.8 GHz personal computer with 2 Gigabytes of RAM running under the Linux operating system. Our algorithm runs in approximately 10 seconds on gels with dimension 5100 x 4200 pixels.

We applied `Contour_Area_Filter` using a threshold of 60,000 for the maximum contour area. Spots above this size were marked as background. In post-processing, we used a pixel size of two for opening (eroding and then dilating the foreground by 2 pixels) and remove any small components with size less than 300 pixels. We used a minimum center size of 300 pixels and a minimum center depth of 3 for segmentation.

We created an annotated image of each master gel, with an identifier marking each spot as described in [18] for the human NotI-EcoRV-HinfI master profile. We compared the spots identified on the annotated master profile with the spots generated by our algorithm. Note that generating spots requires not only identifying foreground spot pixels, but also segmenting the foreground into individual spots. We were interested in measuring the success of the filtering algorithm, not the segmentation one, and so did not count differences in segmentation as added or missing spots.

Analysis of the three profiles described above resulted in excellent correlations between the spots annotated by hand and those identified by the algorithm. Of 2,371 spots, 2,291 spots, and 3,219 spots identified on the annotated human NotI, human AscI and mouse NotI profiles, approximately 99%, 97%, and 96%, respectively, were correctly identified. In addition, the algorithm added small numbers of spots not seen in the annotated images. A spot was only considered added by our algorithm if its pixels did not lie in any of the

spots on the master profile. Similarly, a spot was considered missed by our algorithm if the pixels of that spot were not identified as foreground by our algorithm. Table 1 shows the breakdown of added and missed spots for each profile. Since spots on the master profiles were marked only at their centers as judged by human analysis of the gels, some degree of subjectivity is necessarily a part of the determination of the extent of a spot.

While the hand annotation in each master profile was extensive, we did find distinct spots missed by the annotation. These spots tended to be near the border of the gel where the analysis was much less complete. We occasionally found unannotated spots in the central region of the image. We did not include such unidentified spots in our count of added spots, as these represent errors in the annotation. As described below, we ran the ImageMaster software on our profiles and compared them with our algorithm. We considered a spot unidentified only if it was visually clear and distinct, its intensity matched the intensity of surrounding identified spots and it was also produced by the ImageMaster software.

In order to better understand the nature of the errors made by the algorithm we found that the added spot errors could be broken down into three distinct classifications: “Faint spots”, “Faint noise”, or “Dark noise”. Faint spots were pixels which had the appearance of a spot but were faint and more difficult to detect by hand.(Figure 2.) Some of these were faint only compared to surrounding spots. Others were so faint that they were only visible after applying contrast enhancement to the image. The “Faint spot” classification of added spots therefore do not necessarily represent errors on the part of the algorithm, but may also represent the advantage in detection capability of the algorithm over the

inherently subjective analysis by hand. Added spots marked “Faint noise” were clusters of light pixels which were identified as spots which did not have the shape or appearance of a spot, even after contrast enhancement. Added spots marked “Dark noise” were clusters of dark pixels probably caused by imperfections or physical marks on the gel (Figure 3.). Very few of these appear on our RLGS gels (Table 2.) Both “Faint noise” and “Dark noise” represent errors in the algorithm where marks that are clearly not true spots were added as spots.

Each annotated master profile is used as a reference for all gels with the corresponding enzyme and genome. An identifier for spots on these other gels were sometimes added to the annotated image, even though the spots were extremely faint or did not appear at all on the annotated image (Figure 4.) In particular, the annotated mouse master gel was created by mixing tissue from three different mouse strains and comparing with gels from the individual strains. Unfortunately, we have no record of whether a spot was identified directly from the master gel or indirectly from another gel. Thus we visually checked that each spot which was missed by our algorithm actually did appear in our digitized autoradiograms. We found approximately sixty spots in the annotated human AscI-EcoRV-HinfI profile and fifty spots in the annotated mouse NotI-EcoRV-HinfI profile which we were unable to visually discern in the image, even after trying various levels of image contrast and brightness. We also checked that the ImageMaster software did not produce these spots. We did not include these missing spots in the list of spots missed by our algorithm.

The missed spots are broken down into two categories in Table 3: “Faint” and “Distinct”. The “Faint” classification is defined the same as for above. The “Distinct” classification represents spots that are clearly present in the image but not identified by the algorithm. These two classes represent true errors of lack of identification by the algorithm. Not surprisingly, nearly all of the missed spot errors occurred in the areas of the highest spot density, which can be problematic for the algorithm since it uses the maximum contour area as a parameter. One can view this as the maximum size of a cluster of overlapping spots. If such a cluster has more pixels than this maximum, the algorithm will remove fainter spots until it breaks the cluster apart (Figure 5.) Such high density regions occur near the right edge and the upper left corner of each image. They can also occur in the neighborhood of largely enhanced spots. Each gel contains about a dozen such large spots generated by the repetitive ribosomal DNAs (rDNAs).

The reason for the higher level of missed spots on the mouse profile most likely stems from two factors. First, the mouse master gel is slightly anomalous since it uses a combination of DNA from three mouse strains, FVB, C57/BL6J, and 129/SV (Plass et al. unpublished). Spots corresponding to DNA fragments which appeared in only one or two strains (strain specific polymorphisms) had significantly lower intensity than spots which appeared in all three strains making them more difficult to detect. Second, the density of spots in the mouse gel is higher than the human gels as the mouse gel contains over 3100 identified spots, compared with under 2400 human.

Comparison to 2-dimensional protein gel analysis software

We compared our filtering results with the results from applying the ImageMaster program from Nonlinear Systems. We note that the ImageMaster system is designed and used for protein, not RLGS, gels. Also, the ImageMaster program does not separate background subtraction from spot segmentation in reporting its filtering results. Our configuration of the ImageMaster system could not handle 5100 x 4200 images so we reduced their dimensions to 2550 x 2100.

ImageMaster identifies foreground pixels by comparing the average intensity of k pixels in a neighborhood of a pixel p with the average intensity of a $4k$ pixels on the boundary of a window around p . It uses three significant parameters: sensitivity, window size and noise. Pixel p is classified as foreground if $(I_p - I_s)/I_p > s/10000$, where I_p is the average intensity in the neighborhood of p , value I_s is the average intensity of the $4k$ pixels on the boundary of the window around p , and s is the sensitivity parameter. Higher values of s detect more spots but give more false positives. The noise parameter is the number k of pixels used for the neighborhood of p . It reduces the effect of high frequency noise on the filtering. The window size determines the size of the spots detected. Smaller window sizes detect smaller spots, but fail to detect large saturated spots. We used sensitivity 9500, window size 15x15, and noise 7 on our 2550 x 2100 images.

The results of our comparison are presented in Table 1. As in the comparison of Contour_Area_Filter and the annotated profiles, we are not interested in the segmentation of the spots, only whether they are included in the image foreground. A spot was counted as appearing in a filtered image if its center lay in the foreground area of that image. ImageMaster did quite well compared to Contour_Area_Filter on the human, NotI-

EcoRV-HinfI profile. However, ImageMaster failed to correctly report some of the large saturated spots in this profile which is a glaring error since those spots are so prominent. With larger window sizes, ImageMaster reported those spots, but then missed many of the smaller ones.

For the human, AscI-EcoRV-HinfI and the mouse, NotI-EcoRV-HinfI profiles, ImageMaster missed considerably more spots compared to Contour_Area_Filter. The missed spots were concentrated in the lower left region of the gels where spots were extremely faint. A higher sensitivity number should have been used for the mouse profiles, detecting more spots at the expense of false positives. On the other hand, ImageMaster was already reporting more false positives than Contour_Area_Filter for the other profiles and a higher sensitivity would simply have increased that number. This illustrates the need to modify the ImageMaster parameters based on the gel or even specific regions within the gel. We did try setting sensitivity to 9999, the maximum, for each gel, but this always produced tremendous numbers of spurious spots.

Discussion

We presented an algorithm for filtering foreground of RLGS images based on contour areas in those images. Background intensity varies greatly over these images and some of the spots are extremely faint. Our algorithm reports correctly the foreground for 95% of the spots in those images, with errors concentrated in regions of high spot density. Good algorithm parameters depend upon spot size and density, not spot intensity, and thus do not need to be modified for each gel.

We compared our algorithm to ImageMaster software. In all cases Contour_Area_Filter agreed with the hand annotated gel more than ImageMaster. As importantly, good filtering parameters in Contour_Area_Filter depend upon the spot size and density, not upon the spot intensities. Since spot size and density are consistent between RLGS profiles, we don't modify the parameters for each gel. Good sensitivity values in ImageMaster depend greatly on spot intensity which make them much more gel dependent. Again we emphasize that ImageMaster was designed for protein gels with much darker protein spots and is used here for comparison purposes only.

The major weakness of our algorithm is in areas of high spot density where fainter spots may be obscured by stronger ones. Areas of high spot density are either on the upper-left or the right side of the gels. In other areas, our algorithm gives 99% accuracy.

Postprocessing or hybrid algorithms could perhaps be used to find faint spots in areas of high density.

Segmentation is inherently a much more difficult problem than filtering background from foreground. Ideally, each individual spot should correspond to a different DNA sequence. However, different DNA sequences may migrate to the same or approximately the same location, making differentiation based on the gel image impossible. If DNA sequences migrate to very close locations, it is extremely difficult and subjective to determine whether the resulting image contains one spot or two. We described an extension to Contour_Area_Filter for spot segmentation. However, we are still working on improving this segmentation algorithm and did not present experimental results.

Acknowledgements

This research was supported by NIH grant 3RE01DE13123-02S1. We also thank Dr. Haifeng Wu for use of the Nonlinear Dynamics ImageMaster software in his lab. We thank Dr Li Yu, The Ohio State University, for providing the mouse Master profile for this work.

References

1. Hatada, I., et al., *A genomic scanning method for higher organisms using restriction sites as landmarks*. Proc Natl Acad Sci U S A, 1991. **88**(21): p. 9523-7.
2. Smiraglia, D.J. and C. Plass, *The study of aberrant methylation in cancer via restriction landmark genomic scanning*. Oncogene, 2002. **12**: p. 5414-5426.
3. Appel, R.D., et al., *Melanie II -- a third-generation software package for analysis of two dimensional electrophoresis images: II. Features and user interface*. Electrophoresis, 1977. **18**: p. 2724-2734.
4. Appel, R.D., et al., *Melanie II-- a third generation software package for analysis of two dimensional electrophoresis images: II. Algorithms*. Electrophoresis, 1997. **18**: p. 2735-2748.
5. <http://www.phoretix.com>.
6. <http://expasy.ch/melanie>.
7. <http://gelmatching.inf.fu-berlin.de>.
8. <http://www.bio-rad.com>.
9. <http://www.amershambiosciences.com>.
10. Sugahara, Y., et al., *An automatic image analysis system for RLGs films*. Mamm Genome, 1998. **9**(8): p. 643-51.
11. Takahashi, K., M. Nakazawa, and Y. Watanabe. *DNAinsight: An Image Processing System for 2-D Gel Electrophoresis of Genomic DNA*. in *Genome Inform. Ser. Workshop Genome Inform.* 1997.
12. Takahashi, K., M. Nakazawa, and Y. Watanabe. *DNAinsight: A Web Based Image Processing System for Large Scale RLGs Analysis*. in *Genome Inform. Ser. Workshop Genome Inform.* 2001.
13. Takahashi, K., et al. *Fully-Automated Spot Recognition and Matching Algorithms for 2-D Electrophoretogram of Genomic DNA*. in *Genome Inform Ser Workshop Genome Inform.* 1998.
14. Takahashi, K., et al. *Automated Processing of 2-D Gel Electrophoretograms of Genomic DNA for Hunting Pathogenic DNA Molecular Changes*. in *Genome Inform. Ser. Workshop Genome Inform.* 1999.
15. Sternberg, S.R., *Biomedical Image Processing*. IEEE Computer, 1983: p. 22-34.

16. Okazaki, Y., et al., *An expanded system of restriction landmark genomic scanning (RLGS Ver. 1.8)*. Electrophoresis, 1995. **16**(2): p. 197-202.
17. Smiraglia, D.J., et al., *A New Tool for the Rapid Cloning of Amplified and Hypermethylated Human DNA Sequences from Restriction Landmark Genome Scanning Gels*. Genomics, 1999. **58**(3): p. 254-262.
18. Costello, J.F., et al., *Aberrant CpG island methylation has non-random and tumor type specific patterns*. Nature Genetics, 2000. **25**: p. 132-138.

genome	enzymes	Annot # spots	CAF # added	CAF # missed	IM # added	IM # missed
human	NotI-EcoRV-HinfI	2371	46	34	91	21
human	AscI-EcoRV-HinfI	2291	60	72	76	171
mouse	NotI-EcoRV-HinfI	3219	47	139	11	408

Table 1. Results from filtering RLGS master profiles using Contour_Area_Filter (CAF) and ImageMaster (IM): Number of labeled spots on hand annotated master profiles, number of spots added or missed by Contour_Area_Filter (including post-processing,) and number of spots added or missed by ImageMaster.

genome	enzymes	faint spots	faint noise	dark noise
human	NotI-EcoRV-HinfI	33	3	13
human	AscI-EcoRV-HinfI	13	39	8
mouse	NotI-EcoRV-HinfI	9	38	0

Table 2. Breakdown of spots added by Contour_Area_Filter which are not identified in annotated gels. Number of faint added spots, number of added spots caused by faint noise on the gel, and number of added spots caused by dark noise on the gel.

genome	enzymes	faint	distinct
human	NotI-EcoRV-HinfI	2	32
human	AscI-EcoRV-HinfI	50	22
mouse	NotI-EcoRV-HinfI	69	70

Table 3. Breakdown of spots missed by Contour_Area_Filter which are identified in annotated gels.

Number of missed faint spots and number of missed distinct spots.

Contour_Area_Filter(I, M)

/* I is an array of pixel intensities */

/* M is the contour area threshold */

1. $A \leftarrow \text{Compute_Contour_area}(I)$;
2. For each pixel p do:
3. If $(A[p] > T)$ then mark p as a background pixel;

Compute_Contour_Area(I)

/* I is an array of pixel intensities */

/* Return array A of contour areas */

1. For each pixel p, do
2. $U[p] \leftarrow p$; /* MakeSet(p) */
3. $A[p] \leftarrow 1$;
4. Sort pixels by intensities in decreasing order;
5. For each pixel intensity γ in decreasing order do:
6. For each pixel p with intensity γ do:
7. For each pixel q adjacent to p do:
8. $s \leftarrow \text{FindSet}(q)$;
9. If $(I[p] \leq I[q]$ and $p \neq s)$, then
10. $U[s] \leftarrow p$; /* Union(p,s) */
11. $A[p] \leftarrow A[p] + A[s]$;
12. For each pixel p with intensity γ do:
13. $r \leftarrow \text{FindSet}(p)$;
14. $A[p] \leftarrow A[r]$;
15. Return(A);

FindSet(q)

1. if $(q \neq U[q])$ then
2. $U[q] \leftarrow \text{FindSet}(U[q])$;
3. return($U[q]$);

Figure 1. Contour Area Filter algorithm.

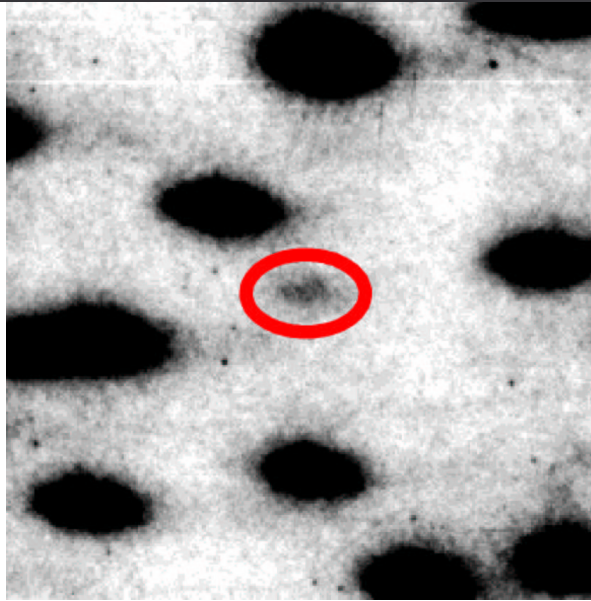


Figure 2. Faint spot added by Contour_Area_Filter.

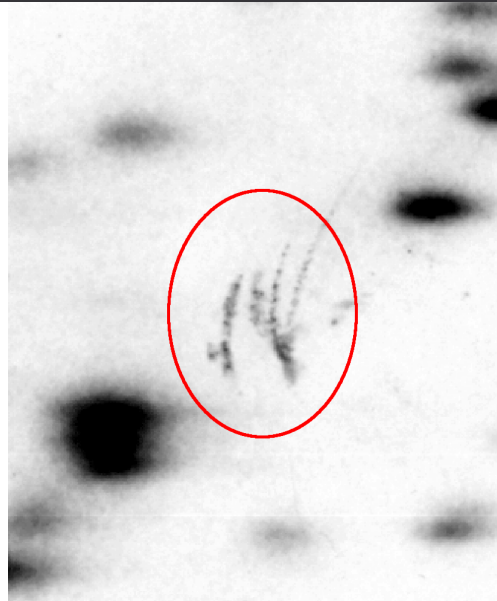


Figure 3. Noise identified as spot by Contour_Area_Filter.

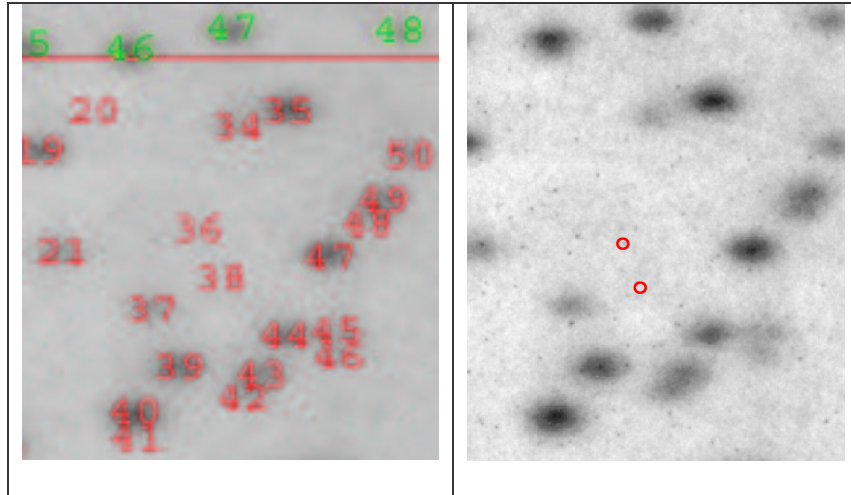


Figure 4. Identified spots 2f:36 and 2f:38 in annotated master mouse gel which are not visible on gel.

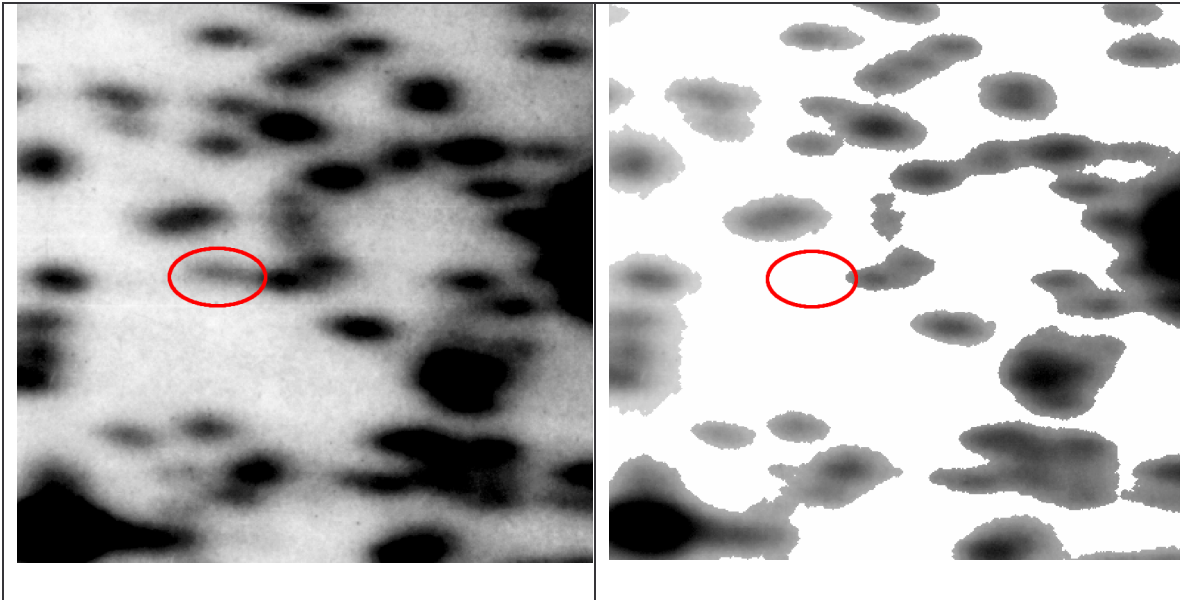


Figure 5. Spot missing from filtered image.

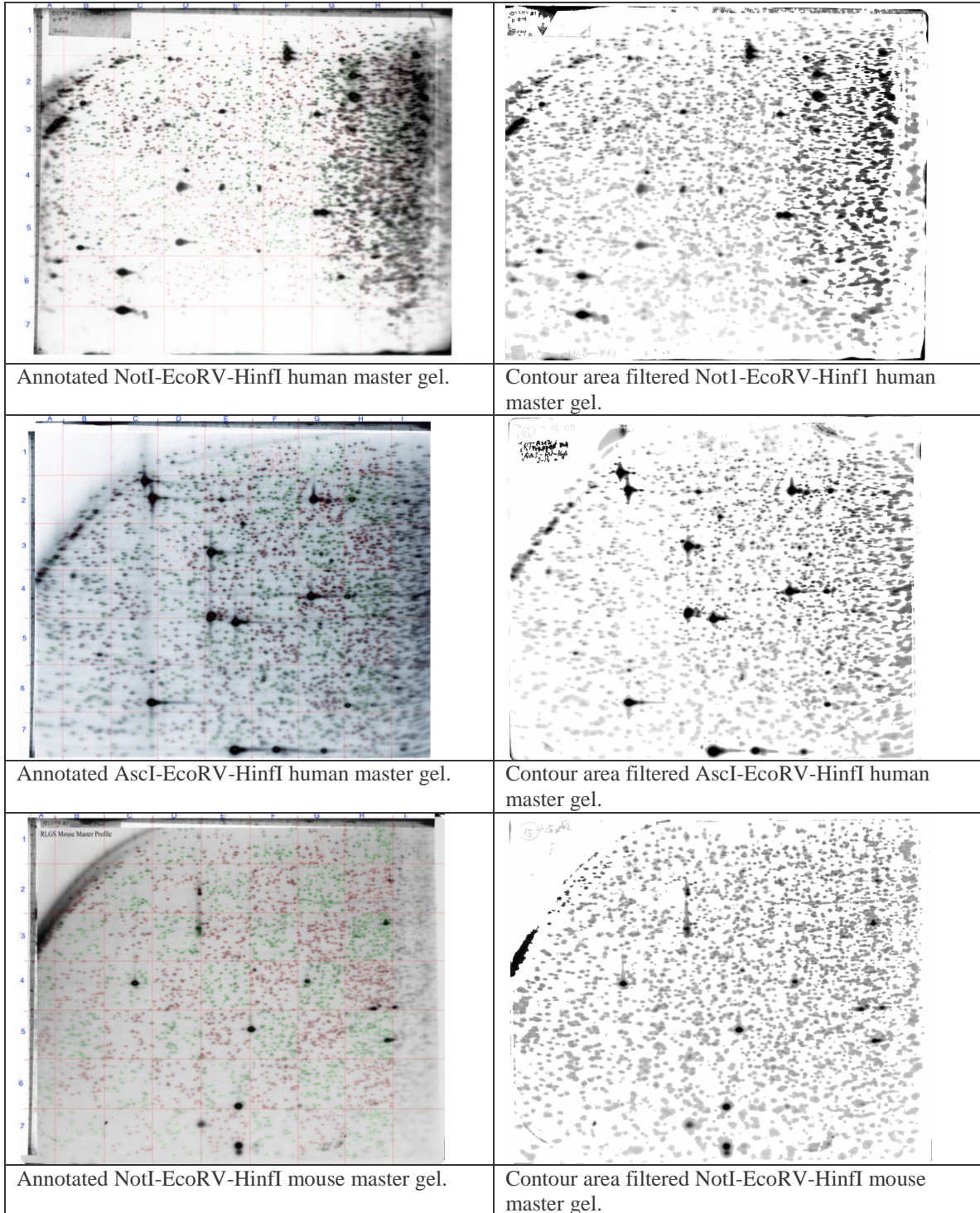


Figure 6. Annotated and contour area filtered master gels.