Polony Identification Using the EM Algorithm Based on a Gaussian Mixture Model

Wei Li*, Paul M. Ruegger†, James Borneman† and Tao Jiang*
*Department of Computer Science and Engineering
University of California, Riverside
Riverside CA 92521
Email: {liw,jiang}@cs.ucr.edu
†Department of Plant Pathology and Microbiology
University of California, Riverside
Riverside CA 92521
Email: paul.ruegger@email.ucr.edu,borneman@ucr.edu

***Abstract*—Polony technology is a low-cost, high-throughput platform employed in several applications such as DNA sequencing, haplotyping and alternative pre-mRNA splicing analysis. Owing to their random placement, however, overlapping polonies occur often and may result in inaccurate or unusable data. Accurately identifying polony positions and sizes is essential for maximizing the quantity and quality of data aquired in an image; however, most existing identification algorithms do not handle overlapping polonies well. In this paper, we present a novel polony identification approach combining both a Gaussian Mixture Model (GMM) and the Expectation-Maximization (EM) algorithm. Experiments on simulated and real images of highly overlapping polonies show that our algorithm has a 10% to 20% increase in recall compared with the existing algorithms, while keeping precision at the same level.**

***Index Terms*—polony identifification; EM; Gaussian Mixture Model;**

I. Introduction

A polony, or “polymerase colony,” consists of a large number of identical copies of a single DNA molecule generated through solid-phase PCR or bridge amplification [5][10]. Since first being developed in 1999 [14], polony technologies have been employed in several important applications, including genotyping and haplotyping [15], alternative pre-mRNA splicing analysis [27] and DNA sequencing [22][21].

In solid-phase PCR experiments, polonies are formed by first mixing sample DNA molecules into a gel matrix and then thinly casting the mixture onto a glass slide. After the gel has hardened, PCR reagents are added to a sealed chamber surrounding the gel and the slide is subjected to thermocyling. During thermocycling, the DNA is exponentially amplified but its lateral movement is somewhat inhibited by the gel. Thus, polonies grow slowly outward and are centered at the randomly-placed DNA molecule from which they originated. Once formed, polonies can be interrogated in various ways depending on the application, but all involve the use of florescence that allow the polonies to be imaged with a laser-scanner or similar device (see the leftmost image of Fig. 1 as an example).

Similar techniques, which use bridge amplification, appear in high-throughput sequencing (HTS) technologies, including Solexa/Illumina, 454 pyrosequencing, SOLiD, etc [13]. HTS technologies can generate 1-2 orders of magnitude more data than traditional Sanger sequencing platforms and do so faster and less expensively [19]. The emergence of such technologies is revolutionizing many sequencing-related research areas such as genome resequencing, SNP discovery, small RNA sequencing, etc [20]. In high-throughput sequencing protocols such as these, the identification of polonies (also called “clusters”) from greyscale images is a required and critical step for gathering data accurately.

Several algorithms have been applied to identify polonies in greyscale images, including edge detection [6][11][16], thresholding [22], watershed segmentation [24][11], etc. In [11], a LoG (Laplacian of Gaussian) filter is applied first for thresholding, followed by a watershed segmentation step to identify potential polonies. Reference [6] detects edges in the image by thresholding the magnitude of the image gradient and then employs a circular Hough Transform to identify circular polonies. Also, [6] uses an exponential function model to calculate polony positions and sizes (see Section II for more details).

The recent HTS technologies often generate hundreds of thousands of greyscale images in a single sequencing project, each of which typically millions of pixels in size. To identify polonies from these images efficiently, “Swift” [25] and “Firecrest” [4] adopt a simple thresholding strategy, together with many pre- and post-processing steps. All of these algorithms work well for isolated polonies (*i.e.*, polonies that do not overlap with each other), as in the case of HTS applications. However, when polony density increases and polonies start to overlap with each other, as in the case of our target application of the polony technology in oligonucleotide fingerprinting of ribosomal RNA genes (OFRG) [23], these algorithms often estimate polony positions inaccurately, and miss dim polonies completely (see Fig. 1).

In this paper, we present a novel Expectation-Maximization (EM) polony identification algorithm based on a Gaussian Mixture Model (GMM). The intensity of a polony in the image

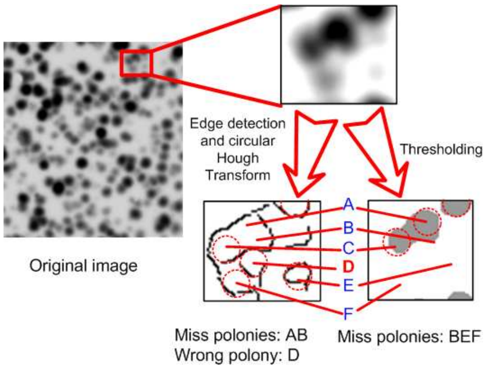

Fig. 1. An image of many overlapping polonies (left), and a zoom-in part of the image (top right). We implemented two different polony identification algorithms published in the literature, and their results on this image are shown in dashed circles (bottom right). The method based on edge detection and the circular Hough transform [6] misses two polonies (A and B) due to their non-circular edges and identifies a spurious polony (D) due to a circular edge formed by neighboring polonies B, C and F. The thresholding method in [22] misses three dim polonies (B, E and F).

is simulated by a Gaussian distribution, and the interaction of multiple polonies is modeled through a GMM. The parameters of the GMM are determined by maximizing a log-likelihood objective function using EM, which is an iterative algorithm for solving maximum likelihood problems with latent variables [7][3]. EM has been widely applied to many areas concerning probabilistic estimations, including image processing [12], haplotype inference [17], speech recognition [8], etc. Experimental results on simulated and real images containing highly overlapping polonies from our OFRG [23] project demonstrate that the GMM-based EM algorithm is able to achieve a higher recall than the existing polony identification methods in the literature while maintaining the same precision.

The rest of this paper is organized as follows. In Section II, we propose the idea of using Gaussian distributions to represent polony intensities, and using EM to maximize the log-likelihood objective function. Section III compares the experimental results of several polony identification methods, while Section IV concludes the paper.

II. Methods

A. The exponential function model of polony intensity

Theoretical analysis [1] and empirical observation [6] of polonies indicate that the shape of a polony can be simulated by an exponential function. As a result, for a polony P positioned at (μ_1, μ_2) in the image and having a radius r, [6] introduces the following function to model the intensity $I(x)$ at pixel $x = (x_1, x_2)$ around P:

$$I(x) = C + A \exp\{-[(x_1 - \mu_1)^2 + (x_2 - \mu_2)^2]/2r^2\} \quad (1)$$

where $C \geq 0$ represents the "background intensity" near P, and $A > 0$ is the "signal intensity" of P. If there are K polonies $P_1, \ldots P_K$ around pixel x, the value of $I(x)$ can be represented as

$$I(x) = C + \sum_{k=1}^{K} A_k \exp\{-[(x_1 - \mu_{k1})^2 + (x_2 - \mu_{k2})^2]/2r_k^2\} \quad (2)$$

where (μ_{k1}, μ_{k2}), r_k and A_k are the center, radius and signal intensity of polony P_k, respectively.

To calculate these parameters, [6] uses an iterative search algorithm to minimize the following sum of square errors between predicted and true intensities:

$$\{C^*, A_k^*, \mu_k^*, r_k^*\} = \min_{C, A_k, \mu_k, r_k} \sum_{m=1}^{M} (I(x^m) - I'(x^m))^2 \quad (3)$$

where there are M pixels around these polonies, each of which x^m has the true intensity value $I'(x^m)$ in the image, and $I(x^m)$ is the predicted intensity using (2). The search algorithm first extracts an initial guess of (μ_{k1}, μ_{k2}) and r_k for all $k = 1, 2, \cdots K$ using edge detection and circular Hough transform, then uses a linear least squares approach to find out C and A_k with fixed values of (μ_{k1}, μ_{k2}) and r_k. After that, (μ_{k1}, μ_{k2}) and r_k are re-evaluated using conjugate gradient descent to minimize (3). Both stages are repeated until convergence or hitting some other stop criteria.

This approach suffers from three drawbacks. First, since two levels of iterations are used, the algorithm converges slowly. Second, the number of polonies (K) must be provided as a parameter before computation; however, if there are many overlapping polonies, it is difficult to accurately estimate K. Finally, the linear least squares approach may assign (nonsensical) negative values to C or A_k.

B. The GMM model and EM algorithm

1) The Gaussian Mixture Model of polony intensity: We use a Gaussian Mixture Model (GMM) to model the intensity distribution of pixels around K polonies $P_1, \cdots P_K$, due to the intuition that the Gaussian distribution has a similar exponential form to (1). The intensity $I(x)$ at pixel x is proportional to the probability density $p(x)$, which in the GMM is a mixture of K Gaussian distributions:

$$\begin{aligned} p(x) &= \sum_{k=1}^{K} P(x \in P_k) P(x | x \in P_k) \\ &= \sum_{k=1}^{K} \pi_k N\left(x | \mu_k, r_k^2 I\right) \end{aligned} \quad (4)$$

where $N(x|\mu_k, r_k^2 I)$ is a two-dimensional Gaussian distribution with mean μ_k and inverse covariance matrix $r_k^2 I$, and $\pi_k = P(x \in P_k)$ is the prior probability that x comes from the kth Gaussian distribution.

The background distribution can be simulated by a Gaussian distribution with a large r value. For a small area around these overlapping polonies, this can be approximated using a uniform distribution U:

$$p(x) = \pi_0 U + \sum_{k=1}^{K} \pi_k N\left(x|\mu_k, r_k^2 I\right) \tag{5}$$

where π_k $(k = 0, 1, \cdots K)$ must satisfy $\pi_k \geq 0$ and $\sum_{k=0}^{K} \pi_k = 1$.

2) EM optimization of the GMM model: Since the intensity of each pixel $I(x)$ is proportional to the probability density $p(x)$, we are able to draw N random samples $X = (x^1, \dots x^n, \cdots x^N)$ from the image according to this probability distribution, where $x^n = (x_1^n, x_2^n)$ is the coordinate of one sample in the image, and the value of N can be set as the sum of all pixel intensities: $N = \sum_{m=1}^{M} I'(x^m)$. The log-likelihood of these N samples is defined as follows:

$$\begin{aligned} &\ln p(X|\{\pi_k, \mu_k, r_k^2\}) \\ &= \sum_{n=1}^{N} \ln \left\{ \pi_0 U + \sum_{k=1}^{K} \pi_k N(x^n|\mu_k, r_k^2 I) \right\} \end{aligned} \tag{6}$$

Maximizing (6) could be easily done using the EM algorithm. The EM algorithm iterates between the **E** step and the **M** step: in the **E** step, the posterior probability of each sample x^n coming from polony P_k is calculated as

$$\begin{aligned} \gamma_{nk} &= P(x^n \in P_k|\pi_k, \mu_k, r_k^2) \\ &= \frac{\pi_k N(x^n|\mu_k, r_k^2 I)}{\pi_0 U + \sum_{k=1}^{K} \pi_k N(x^n|\mu_k, r_k^2 I)} \end{aligned} \tag{7}$$

for $k \neq 0$. For $k = 0$, we have

$$\gamma_{n0} = \frac{\pi_0 U}{\pi_0 U + \sum_{k=1}^{K} \pi_k N(x^n|\mu_k, r_k^2 I)} \tag{8}$$

And in the **M** step, maximizing (6) leads to

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} x^n, k = 1 \cdots K \tag{9}$$

$$\begin{aligned} r_k &= \frac{1}{2N_k} \sum_{n=1}^{N} \gamma_{nk} [(x_1^n - \mu_{k1})^2 + (x_2^n - \mu_{k2})^2], \\ &\quad k = 1 \cdots K \end{aligned} \tag{10}$$

$$\pi_k = \frac{N_k}{N}, k = 0 \cdots K \tag{11}$$

where

$$N_k = \sum_{n=1}^{N} \gamma_{nk}, k = 0 \cdots K \tag{12}$$

3) Preprocessing: The sizes of the image are usually very large (for example 2968×4400 in our experiments), making it impractical for EM input directly. Instead, we use the watershed algorithm [24] to first split the image into smaller fragments, which EM can process more efficiently. If a fragment is too small it is merged into a neighboring fragment. The parameters for the watershed algorithm are set up empirically such that each fragment includes approximately 1-5 polonies. We then pass the initial estimates of the positions and radii of polonies to EM by first applying edge detection and circular Hough transform as described in [6].

4) Choosing the best number of polonies: As the number of polonies (K) increases, the log-likelihood value of (6) will also increase. To choose a proper value of K, we follow [26] to use the *minimum message length* criterion [9]:

$$\begin{aligned} K^* &= \underset{k \in [K_e - D, K_e + D]}{\text{argmax}} L(K = k, X) \\ &= \ln P(X|K = k) - \frac{1}{2} \sum_{i=0}^{k} c_i \log \left\{ \frac{N \pi_i}{12} \right\} \\ &\quad - \frac{k}{2} \log \frac{N}{12} - \frac{\sum_{i=0}^{k} (c_i + 1)}{2} \end{aligned} \tag{13}$$

where $\ln P(X|K = k)$ is the maximum value of log-likelihood function (*i.e.,* (6)) obtained from EM by setting $K = k$, and c_i is the number of free parameters for each Gaussian distribution ($c_i = 1$ for $i = 0$, and $c_i = 3$ for $i \neq 0$).

The range of K is set to $[K_e - D, K_e + D]$, where K_e is the number of polonies identified in the edge detection and circular Hough transform steps, and D is a value chosen by the user.

III. Experiments

A. Real and simulated images

Seven real images from polony PCR experiments in our ongoing OFRG [23] project and 15 simulated images are used to evaluate three different polony identification algorithms. Pertinent statistics of these images, and the images used in [6], are shown in Table I for comparison. Compared with the images in [6], we use images with much higher polony density to evaluate the performance of different algorithms mainly on overlapping polonies. In the seven real images, the polony density is 5-26 times higher than those in [6]; and the simulated images have polonies with densities 17-44 times higher than those in [6].

The polony positions in the seven real images were manually labeled; in the simulated images, the positions of polonies are uniformly distributed. We also add some large "polonies" (whose radii are 30-50 times bigger than those of real polonies) to simulate the variation of background intensities.

TABLE I
Statistics of the images used for evaluation in our experiments and in [6].

Source	Image size	Polony	Density[1]
Images in [6]	1726×2485	about 250	5.8×10^{-5}
Real images	400×400 -1000×1000	300-500	$3.7\text{-}16\times10^{-4}$
Simulated images	600 × 800	500-1300	$1.0\text{-}2.7\times10^{-3}$

[1] Measured in terms of the average number of polonies per pixel.

B. Polony identification

We compare the results of three different polony identification algorithms: Expectation-Maximization (**EM**) proposed before, Edge detection followed by circular Hough Transform (**EHT**) proposed in [6], and a naïve approach of identifying Local-Maximum pixels (**LM**). A local-maximum pixel is defined as a pixel whose intensity is greater than all eight of its neighbor pixels. The results of thresholding [22] and watershed [11] methods are not included here, since they both perform worse than EHT and LM in terms of precision and recall on all of our test images.

The Precision-Recall curve (PR curve) and the Area Under PR curve (AUPR) are used to evaluate and compare each algorithm's ability to identify polonies. The PR curve is drawn from several *precision-recall* value pairs of each algorithm. The *precision* and *recall* of an algorithm are defined as follows: if there are M true polonies, and that algorithm identifies N polonies, K of which are true polonies, then $precision = \frac{K}{N}$, and $recall = \frac{K}{M}$.

The left plot of Fig. 2 shows the PR curve for the highest density real polony image, and the middle and right plots of Fig. 2 show the AUPR values plotted for all 7 real and 15 simulated polony images, respectively. As can be seen from Fig. 2, all three algorithms perform well on low density images (density less than 0.6×10^{-3}), with recall over 80% and precision over 90%. But when polony densities are above 0.6×10^{-3}, EM outperforms the other two approaches decisively.

It may seem surprising that the performance of the naïve LM approach can be so close to the more sophisticated EHT and EM algorithms at lower densities. Perhaps equally surprising is that the LM approach performs as well as EHT at higher densities. The reason for this is that the identifying characteristics of non-overlapping polonies used by the LM and EHT algorithms change during overlap events; local maximum pixels are shifted towards each other and often merge, and polony edges diverge from a perfect circle. The LM approach may fail when a shifted or merged local maximum is no longer a polony center. EHT may fail by identifying a circular edge coincidentally formed by a group of overlapping polonies as a (spurious) polony (such as the polony D in Fig. 1), or overlooking real polonies that induce non-circular edges (such as the polonies A and B in Fig. 1).

EM is superior to both LM and EHT because its underlying assumption - that polonies have intensity profiles which follow a Gaussian distribution - is more valid during overlap events than the assumptions of the other two algorithms due to the nature of polony growth.

All three algorithms are implemented as separate plugins of ImageJ [18][2], a JAVA based open-source image processing tool developed by the National Institutes of Health (NIH). We ran the three algorithms on a laptop with Intel Core2 Duo 2.4 GHz CPU and 1.5 GB memory, and the average processing time per polony is 4ms for LM, 12ms for EHT and 1010ms for EM, respectively. We see that the superior performance of EM does not come without a price: it takes on average 84 times as long as EHT for processing each polony. Each of the complete images in our real data experiments (consisting of up to 2968×4400 pixels) takes approximately 1-2 hours for EM to process.

TABLE II
COMPARISON OF THE EXPONENTIAL FUNCTION MODEL AND THE EM MODEL.

Model	Average time per polony	Average distance	MSE of polony radii
Edge detection & Hough transform	12ms	3.12	7.21
Exponential function & the algorithm in [6]	585ms	2.70	5.55
GMM & fixed-K EM	261ms	2.72	3.85
GMM & automatic-K EM	1010ms	2.68	4.37

C. Parameter estimation

Both the exponential function model (*i.e.*, (1)) and the GMM model (*i.e.*, (5)) are able to estimate polony positions and radii with sub-pixel accuracy. We use 15 simulated images to compare both models. Signal and background estimations are not compared because both models handle them so differently in relation to polony signal (A_i in the exponential function model and π_i in the GMM model) and background intensity (C in the exponential function model and π_0 in the GMM model).

Table II shows the comparison of both models, including the comparison of the average processing time of each polony, the average distance between true and predicted polonies, and the mean squared error (MSE) for polony radii. Table II also includes the result of edge detection and circular Hough transform, which is the initial estimate of polony positions and radii for both EM and the search algorithm in [6]. We use two variations of EM, "fixed K" and "automatic K": the "fixed K" version of EM will assign K=K_e, where K_e is the number of polonies found by edge detection and circular Hough transform. In the "automatic K" version of EM, different K values between $[K_e - 3, K_e + 3]$ are tried and the best K value is selected using the *MML* criterion (see (13)).

From Table II, we see that both models need much more time than the iteration-free edge detection and circular Hough transform approach, but achieve much higher precisions in parameter estimation, especially in estimating polony radii.

Both models achieve similar precisions on polony positions and radii, although the exponential function model estimates polony radii slightly worse than EM. With a fixed value of K, EM only requires a half of the execution time needed by running the search algorithm in [6]. This is because EM has an analytical solution to minimize the objective function (*i.e.*, (6)) with known latent variable (γ). For the exponential function model however, even with fixed A_i and C in (2), minimizing the objective function (*i.e.*, (3)) still needs a conjugate gradient descent loop.

If the number of polonies is unknown, EM needs approximately twice as long as the running time of the search algorithm based on the exponential function model, since EM is executed several times with different K values. However, EM is able to identify more polonies while maintaining the

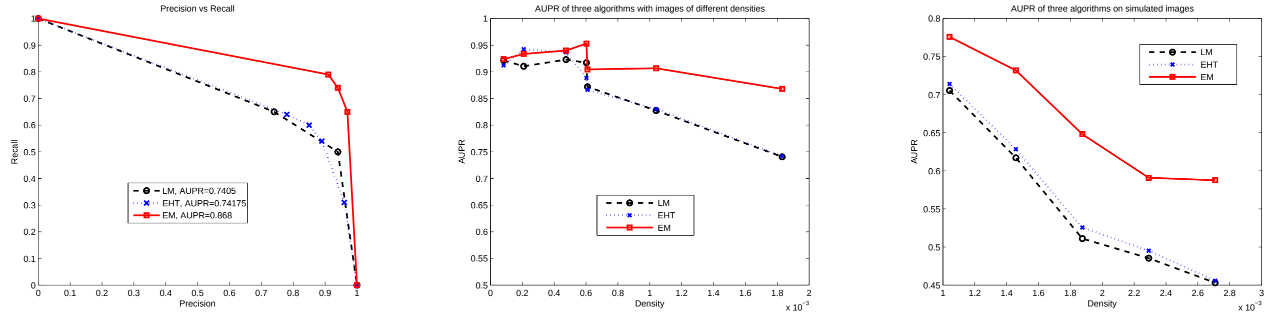

Fig. 2. Precision-Recall curve of the highest density real polony image (left), and the AUPR values plotted for all seven real polony images (middle) and 15 simulated images (right). Plotted results are from the Expectation-Maximization (**EM**), Edge detection/circular Hough Transform (**EHT**) and Local-Maximum (**LM**) algorithms.

same estimated precision of the other two algorithms.

IV. Conclusion

In this paper we use a Gaussian Mixture Model (GMM) to model the interaction of overlapping polonies, and the Expectation-Maximization (EM) algorithm to identify polony positions and sizes. Compared with the previous exponential function model and the search algorithm [6], we show that this approach increases the recall by 10% to 20% while attaining the same level of precision. When the number of polonies (K) is fixed, EM is twice as fast as the search algorithm in [6].

However, EM suffers from its slow execution speed compared with the local-maximum and edge detection/circular Hough transform approaches, especially when the number of polonies needs to be determined computationally. This prevents it from being used in high-throughput sequencing (HTS) images, where hundreds of thousands of images need to be processed quickly. For this reason, [25] uses a simple thresholding strategy to identify polonies, which needs on average only 5 milliseconds processing time for each polony. Note that such a simple method would work well for HTS images because their polonies are generally isolated, but will not be able to handle images from applications such as oligonucleotide fingerprinting of ribosomal RNA genes (OFRG) [23], which is our target application, that may contain many overlapping polonies.

Further improvements in polony identification would be to find a faster and/or more accurate way of determining the number of polonies (K) when overlap occurs, for instance, by leveraging the additional information contained in subsequent images of a base-by-base sequencing reaction.

V. Acknowledgement

The research is supported in part by NIH grant 5R01AI078885. We thank Zhiyao Duan from Northwestern University for his insightful idea of using a GMM, and Jianxing Feng and Yu-Ting Huang for their constructive comments on the manuscript.

References

[1] John Aach and George M. Church. Mathematical models of diffusion-constrained polymerase chain reactions: basis of high-throughput nucleic acid assays and simple self-organizing systems. *Journal of Theoretical Biology*, 228(1):31–46, 2004.

[2] M.D. Abramoff, P.J. Magelhaes, and S.J. Ram. Image Processing with ImageJ. *Biophotonics International*, 11(7):36–42, 2004.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[4] C.G Brown. Solexa/illumina gapipeline product and product documentation.

[5] Adessi C, Matton G, Ayala G, Turcatti G, Mermod J, Mayer P, and Kawashima E. Solid phase dna amplification: characterisation of primer attachment and amplification mechanisms. *Nucleic Acids Research*, 28(87):1–8, 2000.

[6] H.L. Cortes and G. Snyder. An efficient algorithm for multiple polony detection. *5th IEEE International Symposium on Biomedical Imaging (ISBI 2008): From Nano to Macro*, pages 716–719, May 2008.

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[8] V. Digalakis, J.R. Rohlicek, and M. Ostendorf. Ml estimation of a stochastic linear system with the em algorithm and its application to speech recognition. *IEEE Transactions on Speech and Audio Processing*, 1(4):431–442, Oct 1993.

[9] M.A.T. Figueiredo and A.K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002.

[10] Fan Jb, Chee MS, and Gunderson KL. Highly parallel genomic assays. *Nature Reviews of Genetics*, 8:632–644, 2006.

[11] Zhang K, Zhu J, Shendure J, Porreca GJ, Aach JD, Mitra RD, and Church GM. Long-range polony haplotyping of individual human chromosome molecules. *Nature Genetics*, 38(3):382–7, Mar. 2006.

[12] Z. Liang, R.J. Jaszczak, and R.E. Coleman. Parameter estimation of finite mixtures using the em algorithm and information criteria with application to medical image processing. *IEEE Transactions on Nuclear Science*, 39(4):1126–1133, Aug 1992.

[13] Daniel MacLean, Jonathan D. G. Jones, and David J. Studholme. Application of 'next-generation' sequencing technologies to microbial genetics. *Nature Reviews Microbioloy*, 7(4):287–296, April 2009.

[14] RD Mitra and GM Church. In situ localized amplification and contact replication of many individual DNA molecules. *Nucl. Acids Res.*, 27(24):e34–, 1999.

[15] Robi D. Mitra, Vincent L. Butty, Jay Shendure, Benjamin R. Williams, David E. Housman, and George M. Church. Digital genotyping and haplotyping with polymerase colonies. *Proceedings of the National Academy of Sciences of the United States of America*, 100(10):5926–5931, May 2003.

[16] Robi D. Mitra, Vincent L. Butty, Jay Shendure, Benjamin R. Williams, David E. Housman, and George M. Church. Digital genotyping and haplotyping with polymerase colonies. *Proceedings of the National Academy of Sciences of the United States of America*, 100(10):5926–5931, 2003.

[17] Zhaohui S. Qin, Tianhua Niu, and Jun S. Liu. Partition-ligation-expectation-maximization algorithm for haplotype inference with single-nucleotide polymorphisms. *The American Journal of Human Genetics*, 71(5):1242 – 1247, 2002.

[18] W.S. Rasband. ImageJ. U.S. National Institutes of Health, Bethesda, Maryland, USA. http://rsb.info.nih.gov/ij/, 1997-2009.

[19] F. Sanger, S. Nicklen, and A. R. Coulson. Dna sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, 74(12):5463–5467, December 1977.

[20] Jay Shendure and Hanlee Ji. Next-generation dna sequencing. *Nature Biotechology*, 26(10):1135–1146, October 2008.

[21] Jay Shendure, Robi D. Mitra, Chris Varma, and George M. Church. Advanced sequencing technologies: methods and goals. *Nature Reviews Genetics*, 5(5):335–344, May 2004.

[22] Jay Shendure, Gregory J. Porreca, Nikos B. Reppas, Xiaoxia Lin, John P. McCutcheon, Abraham M. Rosenbaum, Michael D. Wang, Kun Zhang, Robi D. Mitra, and George M. Church. Accurate Multiplex Polony Sequencing of an Evolved Bacterial Genome. *Science*, 309(5741):1728–1732, 2005.

[23] Lea Valinsky, Alexandra J Scupham, Gianluca Della Vedova, Zheng Liu, Andres Figueroa, Katechan Jampachaisri, Bei Yin, Elizabeth Bent, Robert Mancini-Jones, James Press, Tao Jiang, and James Borneman. *Molecular Microbial Ecology Manual (2nd Ed)*, chapter Oligonucleotide Fingerprinting of Ribosomal RNA Genes (OFRG), pages 569–585. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004.

[24] Lee Vincent and Pierre Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE PAMI, 1991*, 13(6):583–598, 1991.

[25] Nava Whiteford, Tom Skelly, Christina Curtis, Matt E. Ritchie, Andrea Lohr, Alexander Wait Zaranek, Irina Abnizova, and Clive Brown. Swift: primary data analysis for the Illumina Solexa sequencing platform. *Bioinformatics*, 25(17):2194–2199, 2009.

[26] Baibo Zhang, Changshui Zhang, and Xing Yi. Competitive em algorithm for finite mixture models. *Pattern Recognition*, 37(1):131 – 144, 2004.

[27] Jun Zhu, Jay Shendure, Robi D. Mitra, and George M. Church. Single Molecule Profiling of Alternative Pre-mRNA Splicing. *Science*, 301(5634):836–838, 2003.