Sequential Monte Carlo for probabilistic cell detection in microscopy images

BIOSTAT 882 final project

Tim White

Due April 30th, 2024

Contents

1 Introduction

1.1 Background and literature review

Cell detection is an essential task in microscopy image analysis, as accurate cell segmentation maps are a prerequisite for downstream analysis by biomedical practitioners and researchers. This task was historically performed through manual annotation, but the advent of high-throughput imaging techniques in recent decades has prompted a shift away from this labor-intensive approach and toward automated detection algorithms [1]. Software pipelines such as `CellProfiler` [2] and `ImageJ` [3] are commonly used for routine segmentation due to their relatively simple interfaces, but deep learning methods are the current state of the art among cell detection algorithms, as architectures based on convolutional neural networks (CNNs) have proven to excel at characterizing cells and tissues in microscopy images [4]. However, successful training of these deep networks requires a high volume of manually segmented example images, and their applicability to specific cell or tissue types is limited to those that are represented in the training set [5]. In addition, these models are not well suited for analyzing images that depict dense regions of clustered or overlapping cells, as the point estimates they produce do not capture the inherent ambiguity of cell positions and properties in crowded images [6].

The Bayesian paradigm is a potentially convenient alternative in this setting because it provides calibrated uncertainty estimates for ambiguous images and enables biomedical experts to incorporate prior information based on their domain knowledge. A typical Bayesian approach to this task is to treat the pixel intensities of an image as observed random variables x and the properties of the imaged objects as latent random variables z, and to characterize the posterior distribution $p(z \mid x)$ via Monte Carlo samples or a variational approximation. This is a challenging environment for inference because it is inherently transdimensional — the number of cells in a microscopy image is generally not known a priori, and hence the number of unknown model parameters to infer is itself unknown. As such, most previous attempts to tackle cell detection through a Bayesian lens have relied on transdimensional sampling algorithms. Perhaps the most prominent example is the work of Al-Awadhi et al., who used birth-death Markov chain Monte Carlo (MCMC) [7]. However, the transdimensional proposals required by these algorithms are notoriously difficult to design and may result in slow mixing if they are designed poorly [8].

1.2 Contribution and outline

Motivated by the shortcomings of deterministic software pipelines, CNNs, and transdimensional MCMC algorithms, we propose a novel approach to probabilistic cell detection for crowded fluorescence microscopy images. Our approach, which is based on likelihood-tempered sequential Monte Carlo (SMC) samplers [9], leverages the parallel processing capabilities of modern GPU computing, and it does not require the user to design, or subsequently sample from, transdimensional proposals. The remainder of this report is guided by the following objectives: (1) Determine the necessary components of a realistic Bayesian model of cells in microscopy images and design an efficient SMC sampler to target the posterior distribution induced by this model; (2) assess the statistical and computational performance of our SMC sampler in a simulation study involving synthetic images generated from a simplified version of our Bayesian model; and (3) compare the accuracy and calibration of our sampler's posterior estimates for several relevant cell detection metrics to a popular open-source cell segmentation pipeline based on thresholding and the watershed algorithm [10].

2 Methods

2.1 Notation and Bayesian model

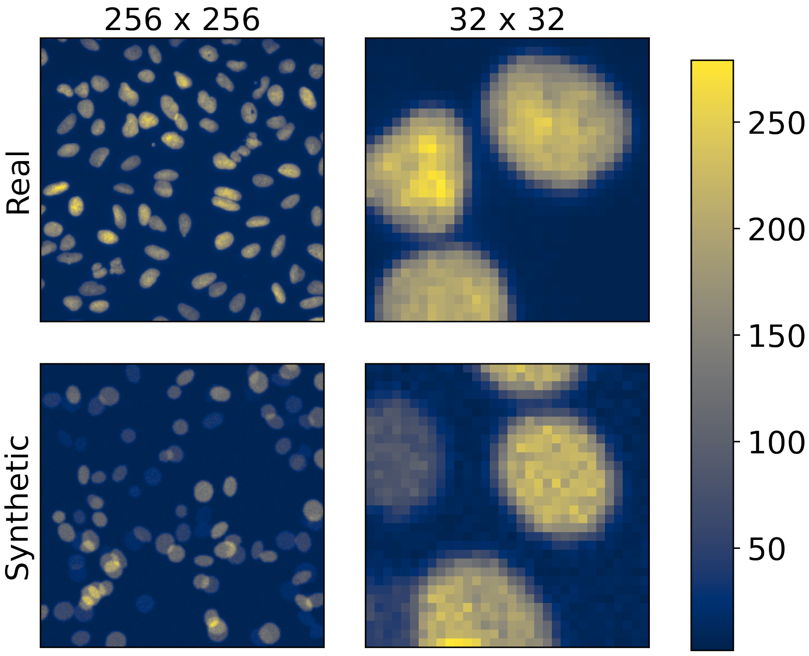
We aim to formulate a model of potentially clustered or overlapping cells in microscopy images that is generic enough to serve as a reasonable baseline for our evaluation of our SMC sampler, but rich enough to provide a realistic approximation of actual fluorescence microscopy images. To make these requirements more concrete, we will design a model that yields images similar to a set of 200 bone cell nuclei images published by the Broad Bioimage Benchmark Collection (image set `BBBC039`) [11]. Our prior and likelihood are heavily informed by previous models of cells in images, namely Al-Awadhi et al. [7] and Lehmussola et al. [12].

Let c denote the number of cells in an image of height H pixels and width W pixels. We will henceforth refer to c as the cell count of the image, and we define $c \sim \text{Uniform}\{0, 1, 2, ..., D\}$ for some maximum number of cells D. Each cell in the image has a location and a fluorescence (i.e., brightness). Given c, we model iid locations $u_1, u_2, ..., u_c \sim \text{Uniform}([0, H] \times [0, W])$ and fluorescence magnitudes $f_1, f_2, ..., f_c \sim \text{Uniform}(f_{\text{min}}, f_{\text{max}})$. The parameters f_{min} and f_{max} depend on the microscope used to capture the images, but are commonly chosen to reflect either an 8-bit scale (with values between 0 and 255) or a 16-bit scale (with values between 0 and 65,535). We used the former in our simulation study since we found that large fluorescence values tended to result in highly peaked likelihoods in practice, which hindered the performance of our sampler.

Cell shapes are irregular in general, but they are typically well modeled by ellipses. As such, we assume that each cell in the image has an angle, a major axis, and a minor axis. Given c, we model iid angles $\theta_1, \theta_2, ..., \theta_c \sim \text{Uniform}(0, \pi)$, major axes $\alpha_1, \alpha_2, ..., \alpha_c \sim \text{Uniform}(\alpha_{\text{min}}, \alpha_{\text{max}})$, and minor axes $\beta_1, \beta_2, ..., \beta_c \sim \text{Uniform}(\beta_{\text{min}}, \beta_{\text{max}})$. The minimum and maximum axis lengths depend on a combination of cell type and image size, and are somewhat arbitrary because images are often downsampled during preprocessing for convenience. In our experiments below, we fix the θ, α, and β of each cell to $\pi/4$, 9, and 6, respectively, for reasons discussed below.

The collection of variables defined above forms a latent variable catalog $z := \{c, \{u_j, f_j, \theta_j, \alpha_j, \beta_j\}_{j=1}^{c}\}$; we will henceforth refer to these catalogs as particles.

Figure 1: Images from `BBBC039` (top) and our Bayesian model (bottom)
(Both 256 × 256 images have a cell count of 102)



Given a particle z, the intensity of the image at pixel (h, w) is $x_{hw} \mid z \sim \text{Poisson}(\lambda_{hw}(z) + \gamma)$, where γ is the constant background fluorescence of the image and $\lambda_{hw}(z) = \sum_{j=1}^{c} f_j \varphi(h, w, z)$. The function $\varphi(\cdot)$ is the point-spread function (PSF), a deterministic function that describes the response of the microscope to the impulses of nearby cells at each pixel. In our simulations, we model the PSF as bivariate Gaussian convolution kernel, and we set the background fluorescence to 10.

We illustrate the realism of this generative model in Figure 1, where we display a 256×256 pixel image generated from our full model (including random ellipse shape parameters) next to a downsampled image of the same size from the `BBBC039` data set. We also display 32×32 pixel cutouts from each image to demonstrate their similarities on a more granular level. Our model is parametric and thus is unable to capture all the nuances of the `BBBC039` cell shapes, but it otherwise provides a close approximation.

2.2 Sequential Monte Carlo samplers for cell detection

Given an image $x = \{\{x_{hw}\}_{h=1}^{H}\}_{w=1}^{W}$, we use a sequential Monte Carlo (SMC) sampler to characterize the posterior distribution $p(z \mid x)$ of possible particles (i.e., collections of cells) that explain the image. As is typical in the Bayesian regime, we assume that $p(z \mid x)$ cannot be sampled from directly due to the intractable normalizing constant $p(x)$, but that the unnormalized joint density $p(x, z) = p(z)p(x \mid z)$ can be evaluated pointwise. Our SMC sampler uses likelihood tempering to approximate a sequence of distributions that gradually progresses from the prior $p(z)$ to the posterior. This sequence takes the form $p(z)p(x \mid z)^{\tau_t}$, and it progresses according to a tempering schedule $0 = \tau_0 < \tau_1 < \cdots < \tau_T = 1$. Likelihood-tempered SMC samplers of this form involve four core procedures: sampling particles from $p(z)$, mutating them with Markov chain Monte Carlo (MCMC) moves, updating their weights to track the above sequence of target distributions, and resampling them to avoid weight degeneracy.

Our sampler involves a methodological trick that renders it a convenient method for transdimensional inference. As mentioned in section 1, it does not require the user to define, or subsequently sample from, transdimensional proposals. Instead, it initializes an equal number of particles with each candidate value of c and preserves the cell count of each particle for the duration of the algorithm. A set of importance weights is maintained for each "block" of particles that have the same cell count, and the resampling step of the algorithm is performed within these blocks using the intra-block weights. The sampler otherwise progresses through the usual mutation and reweighting stages of SMC, exploring the space of positions and properties for particles with various cell counts and iteratively assigning weights to catalogs based on their plausability under the current target. A separate set of inter-block importance weights is also maintained throughout the algorithm, and the inter-block weights of the particles returned after the final iteration can be used to assess the posterior probabilities of different cell counts c.

We formalize this procedure in Algorithm 1. For the mutation step, we use a k-step random walk Metropolis-Hastings kernel that is invariant under the current target distribution, as this is the default MCMC kernel recommended for many SMC samplers. For the tempering and resampling steps, we use typical procedures suggested in the SMC literature; we refer the reader to [9] for more details.

3 Simulation studies

3.1 Experiment settings

We evaluate the cell detection capabilities of our SMC sampler using 1,000 synthetic images of size 32 pixels by 32 pixels with cell counts ranging uniformly from zero to four. We generate these images using a simplified version of our Bayesian model in which the angles and axes of the cells are fixed at the values mentioned in subsection 2.1. In exploratory runs of our SMC sampler on the full version of the model, we found that the computational burden of inferring fluorescences, locations, and cell shapes was prohibitive. We encountered similar computational constraints when we ran the sampler on larger images with more cells. This is a limitation of the sampler in its current form. We theorize that this problem could be alleviated with more complex, component-wise MCMC mutation kernels, as opposed to the random walk kernel that we used here.

We run our SMC sampler once for each image and record the posterior mean source count and posterior mean total fluorescence. We compare the performance of our approach to a popular deterministic cell segmentation pipeline that comprises two steps: thresholding and the watershed algorithm [10]. We implement this

Algorithm 1 `SMC sampler, stratified by cell count`

```
Input: image $x$; prior $p(z)$; likelihood $p(x \mid z)$; MCMC kernel $M_\tau(z, dz)$ for $\tau \in [0,1]$;
       number of blocks $B$ (indexed by $b$); number of particles per block $N$ (indexed by $n$);
       choice of resampling scheme (e.g., multinomial, stratified, systematic).

  Iteration $t \leftarrow -1$. Tempering exponent $\tau_t \leftarrow 0$.
  while $\tau_t < 1$ do
    $t \leftarrow t + 1$.
    if $t = 0$ then
      Particles $z_t^{bn} \sim p(z)$ such that $\forall b,\ c_t^{b1} = \cdots = c_t^{bN}$.
      Unnormalized weights $w_t^{bn} \leftarrow 1$.                                   } Initialize
      Intra-block normalized weights $\widetilde{W}_t^{bn} \leftarrow \frac{1}{N}$.
      Inter-block normalized weights $W_t^{bn} \leftarrow \frac{1}{BN}$.
    if $t > 0$ then
      for block $b \in \{1, ..., B\}$ do
        Resample $N$ indices $\{A_t^{bn}\}_{n=1}^N \leftarrow \texttt{resample}(N,\ \{w_{t-1}^{bn}\}_{n=1}^N)$.      } Resample
        Reset $\widetilde{W}_{t-1}^{bn} \leftarrow \frac{1}{N}$ and $W_{t-1}^{bn} \leftarrow \frac{1}{N}\sum_{n=1}^N W_{t-1}^{bn}$.
    Mutate $z_t^{bn}$ with $k$-step kernel $M_{\tau_{t-1}}(z_{t-1}^{A_t^{bn}}, dz)$.      } Mutate
    Update $\tau_t \leftarrow \tau_{t-1} + \delta$, where $\delta \in [0, 1-\tau_{t-1}]$.      } Temper
    Update $w_t^{bn} \leftarrow W_{t-1}^{bn} p(x \mid z_t^{bn})^{\tau_t - \tau_{t-1}}$.
    Update $\widetilde{W}_t^{bn} \leftarrow w_t^{bn} / \sum_n w_t^{bn}$.                 } Update weights
    Update $W_t^{bn} \leftarrow w_t^{bn} / \sum_b \sum_n w_t^{bn}$.
Output: Weighted particle approximation $\{\{z_t^{bn}, W_t^{bn}\}_{b=1}^B\}_{n=1}^N$ of $p(z \mid x)$.
```

deterministic procedure (which we will henceforth refer to as "watershed") using the open-source Python library `scikit-image`. Our SMC sampler is implemented in PyTorch, and we ran these experiments on one NVIDIA GPU.[1]

3.2 Results

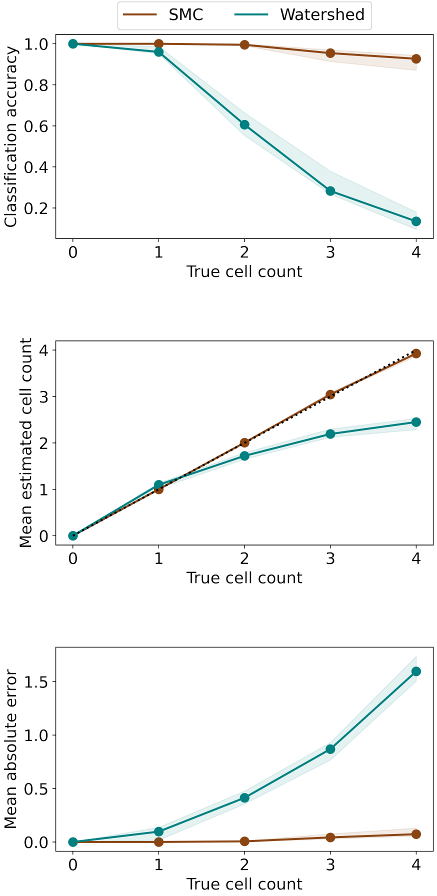
Our SMC sampler yields a correct estimate of the true cell count (i.e., posterior mean count rounded to the nearest integer) in 977 of the 1,000 images (97.7%), with a mean absolute error of 0.023. The watershed algorithm does not achieve the same level of accuracy, as it yields a correct estimate in only 613 of the 1,000 images (61.3%) with a mean absolute error of 0.564. Figure 2 displays the classification accuracy, calibration, and mean absolute error of the two methods among images with the same true cell count. We observe that the performance of the watershed algorithm degrades in more crowded images, while our SMC sampler maintains strong performance even in images with four cells.

Figure 3 and Figure 4 serve as a sanity check for our SMC sampler. If our algorithm produces high-quality weighted particle approximations of the posterior

[1] Code is available at `https://github.com/timwhite0/smc_object_detection/tree/cells`.

Figure 2: Accuracy, calibration, and MAE of estimated cell counts

(Error bands are 90% bootstrap percentile intervals)



distribution for a given image, then it should be capable of characterizing the fluorescences and locations of the cells in that image. Figure 3 demonstrates that SMC's reconstructions of our synthetic images provide a close (and noiseless) approximation to the original images, while Figure 4 illustrates that the sampler's estimates of the posterior mean total fluorescence are well calibrated.

4 Discussion

The results of our experiments demonstrate that our SMC sampler is capable of detecting cells in noisy, pixelated images and inferring a posterior distribution over their latent properties. However, these experiments involved small images with a relatively small number of homogeneous cells, and therefore our sampler would likely require some methodological tweaks to achieve the same performance on a real data set such as `BBBC039`. One potential improvement would be to incorporate Gibbs-like moves into the MCMC kernel in the mutation step, as this would allow moves to be proposed for each latent property iteratively. Another improvement would be to scale the algorithm to large images using a tiling scheme

Figure 3: SMC reconstructions of four example images from our model

(SMC reconstructions are created using the particle with the highest weight)

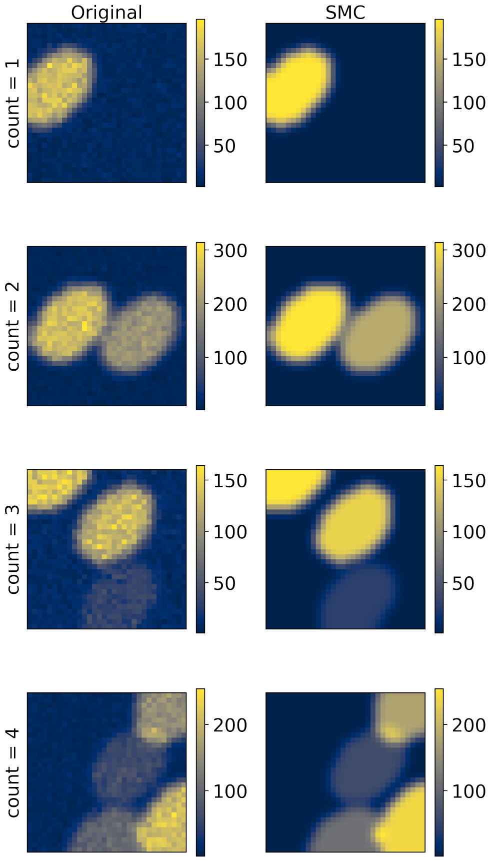
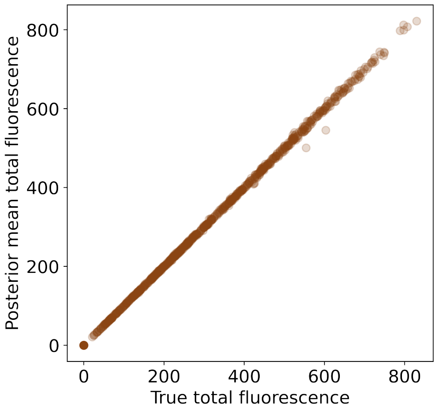


Figure 4: True total fluorescence vs. posterior mean estimated by SMC

(Each dot represents one of the 1,000 images)



in which the posterior samples from many tiles are combined using product-form estimators [13]. Finally, it is of interest to apply our SMC algorithm to more challenging microscopy tasks, such as the detection of malaria-infected cells or cancerous cervical cells.

5 References

[1] Roy Wollman and Nico Stuurman. “High throughput microscopy: from raw images to discoveries”. In: *Journal of Cell Science* 120.21 (2007), pp. 3715–3722.

[2] Anne E Carpenter et al. “CellProfiler: image analysis software for identifying and quantifying cell phenotypes”. In: *Genome Biology* 7 (2006), pp. 1–11.

[3] Caroline A Schneider, Wayne S Rasband, and Kevin W Eliceiri. “NIH Image to ImageJ: 25 years of image analysis”. In: *Nature Methods* 9.7 (2012), pp. 671–675.

[4] Erick Moen et al. “Deep learning for cellular image analysis”. In: *Nature Methods* 16.12 (2019), pp. 1233–1246.

[5] Noah F Greenwald et al. “Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning”. In: *Nature Biotechnology* 40.4 (2022), pp. 555–565.

[6] Fuyong Xing and Lin Yang. “Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: a comprehensive review”. In: *IEEE Reviews in Biomedical Engineering* 9 (2016), pp. 234–263.

[7] Fahimah Al-Awadhi, Merrilee Hurn, and Christopher Jennison. “Three-dimensional Bayesian image analysis and confocal microscopy”. In: *Journal of Applied Statistics* 38.1 (2011), pp. 29–46.

[8] Peter J. Green. “Reversible jump Markov chain Monte Carlo computation and Bayesian model determination”. In: *Biometrika* 82.4 (1995), pp. 711–732.

[9] Nicolas Chopin, Omiros Papaspiliopoulos, et al. *An Introduction to Sequential Monte Carlo.* Springer, 2020.

[10] Norberto Malpica et al. “Applying watershed algorithms to the segmentation of clustered nuclei”. In: *Cytometry: The Journal of the International Society for Analytical Cytology* 28.4 (1997), pp. 289–297.

[11] Vebjorn Ljosa, Katherine L Sokolnicki, and Anne E Carpenter. “Annotated high-throughput microscopy image sets for validation.” In: *Nature Methods* 9.7 (2012), pp. 637–637.

[12] Antti Lehmussola et al. “Synthetic images of high-throughput microscopy for validation of image analysis methods”. In: *Proceedings of the IEEE* 96.8 (2008), pp. 1348–1360.

[13] Juan Kuntz, Francesca R Crucinio, and Adam M Johansen. “Product-form estimators: exploiting independence to scale up Monte Carlo”. In: *Statistics and Computing* 32.1 (2022), p. 12.