This CVPR workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Theia: Bleed-Through Estimation with Convolutional Neural Networks

Najib Ishaq
najib.ishaq@nih.gov

Nathan Hotaling
nathan.hotaling@nih.gov

Nicholas Schaub
nick.schaub@nih.gov

Information Technology Resources Branch
National Center for Advancing Translational Sciences
National Institutes of Health

Axle Research And Technologies

Abstract

Microscopy is ubiquitous in biological research, and with high content screening there is a need to analyze images at scale. High content screening often uses multichannel, epifluorescence microscopy (multiplexing), and fluorescent images often exhibit channel mixing, or bleed-through effects, which need to be corrected before subsequent analysis (e.g. segmentation, feature extraction, etc). In this paper we present Theia, an algorithm for bleed-through correction that requires little to no a priori information about the source or content of the images (i.e. number of channels). Theia uses a novel neural network architecture inspired by Siamese Networks and Least Absolute Shrinkage and Selection Operator (LASSO) regression to learn convolutional filters that remove bleed-through. We use metrics for quantifying bleed-through, and show Theia exhibits good capacity for removing bleed-through on both synthetic and real fluorescent images. Theia was benchmarked to demonstrate scalability across diverse datasets with varying degrees of bleed-through and numbers of channels. Since Theia learns a set of convolutional kernels using popular neural network frameworks, it can make use of GPU acceleration when scaling to large datasets.

1. Introduction

Multiplexed fluorescent microscopy has enabled significant advances in biology and medicine, permitting spatial localization of cellular structures. This technique produces images with high signal-to-noise ratios, providing more accurate visualizations of proteins, lipids, DNA, and other biological molecules. Multiplexing enables one to visualize the physical location of multiple molecules simultaneously (in separate channels), and is often used to show where molecules might associate with each other. Different molecules can be tagged with different fluorophores, so overlapping fluorescence indicates interactions between those molecules. However, images from such microscopes exhibit bleed-through effects, where signal from one channel can be seen in adjacent channels [14].

Figure 1A shows excitation and emission spectra of different fluorophores. Bleed-through occurs when multiple fluorophores have overlapping excitation and emission spectra, permitting multiple fluorophores to be excited by the same light and allowing multiple fluorophores to pass through an emission filter. This results in multiple signals being captured in a single image, where undesired signals are said to be "bleeding through" the filter and become detectable in the image. The three fluorophores shown demonstrate a case where the spectra and their associated filters have significant overlap with each other. In this case, when excitation light is projected onto a sample, multiple fluorophores are excited, although with different magnitudes, causing bleed-through and producing overlapping signals in captured images [23, 24]. Therefore, bleed through is a function of spectral overlap, wide emission filters, and intensity of overlapping signal.

Bleed-through can be mitigated, in part, by careful selection of fluorophores that have minimal overlap in their excitation and emission spectra and by a careful selection of the excitation and emission filters. However, this is not always possible and even when it is, minuscule amounts of overlap among the spectra can still cause significant bleed-through between channels. Recent work found that bleed-through

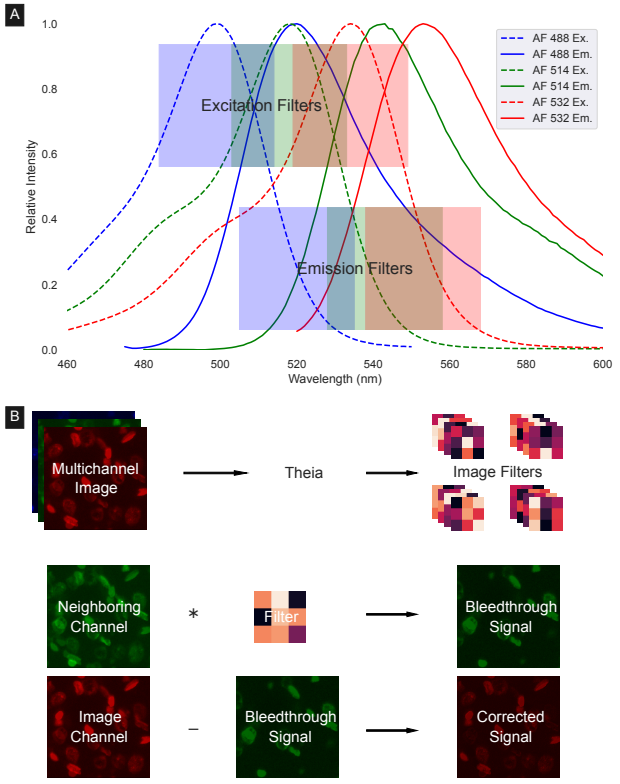

Figure 1. **A visual explanation of bleed-through and overview of Theia.** Panel A shows the excitation spectra (dotted lines) and the emission spectra (solid lines) of three fluorophores. The three excitation filters are shown in the top row as three translucent rectangles, and three emission filters are shown in the bottom row. Light passes through one excitation filter, and light from excited fluorophores passes through the corresponding emission filter. Panel B gives an overview of how Theia performs bleed-through correction. The first row shows how a multichannel image is used by Theia to learn a set of image filters (convolutional kernels) for each channel. The middle row shows how learned filters are used to extract bleed-through signal from an image. The resulting bleed-through signal is subtracted from the original signal (bottom row).

levels as low as 2% are enough to contaminate results [11].

Bleed-through correction is an important early step in workflows that analyze fluorescent signal in biomedical images. Examples include quantification of relative fluorescence intensity to assess changes in a signal within a region of interest, segmentation of structures in different channels [18], and evaluation of the correlation among structures from different channels. If not dealt with correctly, bleed-through artifacts cause problems such as an increased false-positive rate in classification and identification algorithms, and artificial increases in the correlation between objects that were targeted with different fluorophores.

Previous approaches to correct for bleed-through in fluorescent images have used various machine learning algorithms, [11, 13, 22] ranging from clustering to generalized linear models. These approaches often (1) require *a priori* information about number of channels, (2) need information/annotations on the content being imaged, (3) do not account for spatial shifts in signal due to optical aberrations, and/or (4) are not designed for scalability in high throughput situations. These gaps can lead to time consuming configuration, the need for domain expertise that are not always available, an inability to analyze large data, extremely lengthy processing pipelines, and/or a less than optimal bleed-through estimation.

In this manuscript, we introduce a simple, scalable, and performant algorithm for bleed-through correction of multichannel images. We call this algorithm Theia, and it addresses the current gaps by: (1) requiring nominal *a priori* knowledge of how the samples were collected, (2) accepting arbitrary number of channels, (3) performing chunked processing to operate on arbitrarily large images, and (4) leveraging parallelized computation on GPUs for fast execution. To help others use Theia, we have packaged the algorithm for use in the following formats: a command-line tool and Python package via GitHub[1] and PyPI[2], a docker container[3], and a plugin for the Web Image Processing Pipelines (WIPP)[4]

2. Related Works

Algorithms for bleed-through correction are an active area of research, and some state-of-the-art approaches are presented in this section. Theia is benchmarked, in Section 4, against those with an open-source implementation.

Least Absolute Shrinkage and Selection Operator

Generalized Linear Models (GLMs), and LASSO in particular, have long been used for bleed-through correction. In [12], the authors used LASSO to find a sparse linear combination of channels that removed the most bleed-through from each channel of a large ($42,906 \times 29,286$ pixels) image. The authors provide an open-source implementation in Python, and while the method performs as expected, it has a human-in-the-loop element in the requirement of a manually annotated Region of Interest (RoI) in the image. This RoI needs to contain background, auto-fluorescence, and the fluorescence signal of interest; thus necessitating careful manual inspection. Additionally, the proposed algo-

[1] https://github.com/polusai/theia
[2] https://pypi.org/project/theia-py/
[3] https://hub.docker.com/r/polusai/bleed-through-estimation-plugin
[4] https://github.com/PolusAI/polus-plugins/tree/master/regression/polus-bleed-through-estimation-plugin

rithm is computationally intensive and took approximately 6 hours on a 32-core CPU running at 3.10 GHz with 256 GB of RAM [12].

The requirements of careful manual annotation of RoIs and the prohibitive computational cost render this model a poor fit for large image datasets and applications where automation is desired. Further, an application of the vanilla LASSO algorithm is susceptible to false-positives when the same object is tagged with multiple fluorophores. In such cases, LASSO can remove such objects from both channels.

k-Means Clustering to Label Fluorophores

In [13], the authors describe an unsupervised algorithm, Learning Unsupervised Means of Spectra (LUMoS), provided as an ImageJ plugin, to learn and separate the spectral signatures of fluorophores. The authors assume that fluorophores are spatially separated, i.e. each pixel is occupied by exactly one fluorophore. LUMoS first normalizes the intensity at each pixel across all channels and then uses k-means++ [2] to approximate each spectral signature in the image as one of the k clusters. For the k-means algorithm, the authors assume one spectral signature for each of the fluorophores and one additional signature for the background. Each pixel is assigned to one of k spectral signatures. Finally, LUMoS creates a new image with k channels, with one channel for pixels in each spectral signature.

By working with images in which fluorophores are not colocalized, LUMoS was able to identify the signal of each fluorophore even when the number of fluorophores was larger than the number of measurement channels. The authors demonstrated this by separating the signals of six fluorophores from an image with four channels. However, this method assumes that fluorophores are spatially separated, and requires knowledge of the number of fluorophores.

Linear Unmixing by Solving the Linear Least Squares Problem

As described in [22], bleed-through correction can also be formulated as a linear unmixing problem. In this work, model used *a priori* knowledge about which fluorophores (and their subsequent emission spectra) the specimens were labelled with to enable them to correct for bleed through and distinguish between a large number of spectrally near fluorophores. By recording the emission spectra at multiple excitation wavelengths, they increased the number of observations in the linear system, thus increasing the number of separable fluorophores.

However, they also labelled each organism with exactly two fluorophores, adding a constraint to the linear model, which allows for more robust solutions. This model, while effective within the above bounds, was limited by the significant amount of required information on the fluorophores, and the strict limitations on the experimental design. Also, this approach became prohibitively computationally expensive for larger images, and images with many objects.

Independent Component Analysis (ICA) for Separating Spectra

Spectral unmixing is particularly ameanble to being formulated as a Blind Source Separation problem and lends itself well to an application of ICA [10, 20].

In [20], the authors used ICA to separate dyes in multichannel fluorescent microscopy images. In strictly controlled scenarios where fluorophores are either spatially separated or the fluorophores have discrete overlap with sharp boundaries, this method demonstrates good signal separation and converges in a small number of iterations. However, in the case of continuous spatial overlap of the fluorophores with smooth gradients, the method requires at least two orders of magnitude as many iterations and even then, it converges to a much higher error value.

While ICA can perform well at spectral unmixing, it has some intrinsic limitations. ICA requires that the number of measured channels to be at least as large as the number of fluorophores [10]. Further, since ICA requires matrix inversions and decompositions, it requires that all images and channels be loaded into memory at the same time, thus severely limiting its feasibility on large images such as [12]. In addition, the Independent Components (ICs) identified by ICA are not guaranteed to be in any particular order, and thus ICA erases any traceability from ICs, i.e. the separated fluorophores, to the original measured channels in the image, thus needing experts to identify what new "channels" are associated with which fluorophore.

3. Methods

Theia is an algorithm for learning convolutional kernels that extract bleed-through signal. A brief overview of the method is shown in Figure 1B, and discussed below. Theia requires minimal *a priori* knowledge of fluorphore spectra It works with multi-channel images and does so scalably both in terms of computational efficiency and the sizes of the images involved.

Theia draws inspiration from Least Absolute Shrinkage and Selection Operator (LASSO) [21] and Siamese Neural Networks [4, 6, 8, 19]. Theia generalizes Siamese Networks to use an arbitrary number of arms, each of which learns a convolutional kernel to correct for bleed-through and spatial shifts by minimizing an objective function that is mathematically identical to the LASSO objective, i.e. generalized linear regression with an $l1$-norm penalty on the coefficients. We call this generalization of Siamese networks a Dynamically Branched Neural Network (DBNN) because the neural network architecture changes as a function of the number of input channels and channel overlap.

Dynamically Branched Neural Networks

Theia relies on a novel neural network architecture which is a generalization of Siamese Networks and is implemented using Tensorflow [1] and Keras [7]. Theia simultaneously learns a number of DBNNs, with each DBNN learning the convolutional kernels required to minimize bleed-through in a single "target" channel. It is dynamically branched in that the number of arms of the network are dependent on the number of channels in the and the amount of channel overlap to consider when learning bleed-through kernels. Figure 2 shows an example of one such DBNN, targeting channel 2 in a multi-channel image, assuming that bleed-through is present only from up to one adjacent channel on each side of the wavelength spectrum. Similar networks can be trained with an arbitrary number of total channels and an arbitrary number of bleed-through contributing channels. The DBNN has a number of arms, each with its own independent convolutional network (in our case, this consists of a single convolutional layer with a one-channel filter). Each arm is responsible for applying a filter to an adjacent channel (i.e. a channel that may contribute bleed-through to the target channel), or to an interaction term between the target and an adjacent channel. Each arm passes on the filtered signal, along with the convolutional filter, to an Aggregation Layer. The aggregation layer computes and applies the LASSO loss to the DBNN. Thus, Theia learns the convolutional kernels required to bleed-through in multi-channel images.

For a target channel C_i and a channel overlap of o, we compute interaction terms with each contributing channel: $I_i^j = \sqrt{C_i \cdot C_j}$ where $j \in \{i-o, i-o+1, ..., i+o-1, i+o\}/\{i\}$. Each contributing channel and each interaction term then passes through a convolutional layer with a single trainable kernel, producing contribution estimations C_i^j and interaction estimations I_i^j, along with corresponding kernels kC_i^j and kI_i^j. These, and C_i, are all then passed to an Aggregation layer A_i, the purpose of which is to apply the LASSO loss to the DBNN. This loss is given by:

$$\frac{1}{|p|}\sum_{p}\left(c_i - \sum_j C_i^j - \sum_j I_i^j\right)^2 + \lambda\frac{1}{|k|}\sum_{k}\left(\left|kC_i^j\right| + \left|kI_i^j\right|\right) \tag{1}$$

where λ is the relative magnitude of the $l1$-penalty, p is the set of pixels and k is the set of kernel coefficients. Equation 1 is equivalent to the LASSO objective, as described in the following subsection.

LASSO

The Least Absolute Shrinkage and Selection Operator (LASSO) (from *scikit-learn* [15]) is a generalized linear model with an $l1$ penalty on its coefficients. The $l1$ penalty produces a sparse set of coefficients, which is important to help prevent false positives because some fluorophores may not have overlapping spectra and cannot bleed into each other. The objective of LASSO, as implemented in *scikit-learn*, is to minimize:

$$\sum_{i=1}^{N}\left(y_i - \sum_j \beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| \tag{2}$$

where the x_{ij}'s are the standardized predictors, the y_i's are the centered response values for $i = 1..N$ and $j = 1..p$, the β_j's are the coefficients to be learned, and λ is a tuning parameter that controls the relative strength of the $l1$ penalty.

To predict the bleed-through coming into a target channel, the pixels for the target channel are the response values and corresponding pixels from neighboring channels are the predictors. The learned coefficients are the factors by which neighboring channels bleed into the target channel. Intuitively, the LASSO objective, in 2, is minimized when all bleed-through has been subtracted away. This method is common in literature [11–13, 22] and was used in [17].

Theia builds upon this with some novel improvements. First, it intelligently selects a small subset of the data which is most likely to contain bleed-through. Second, it uses interaction terms which helps to preserve the real signal espe-

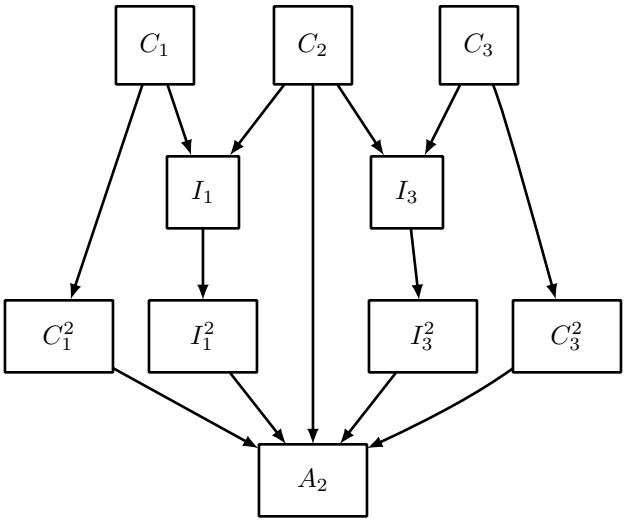

Figure 2. **A DBNN architecture from Theia.** The Dynamically Branched Neural Network architecture that has a number of arms proportional to the number of input channels. In this figure, three channels are shown (top row, $C_1 - 3$), and the bleed-through from C_1 and C_3 into C_2 is learned. All convolutional layers consist of one trainable kernel and zero bias. Interaction terms are calculated (I_1, I_3), then convolutional layers are applied to each channel and interaction term (C_1^2, C_3^2, I_1^2, I_3^2). Outputs from the convolutional layer are then combined in the aggregation layer (A_2) and the loss is back-propagated.

cially in the case of colocalized fluorophores.

Given any fixed set of fluorophores and emission filters, the brightest regions in a channel will bleed through the most into other channels. Thus, Theia selects the brightest patches (tiles) from each channel. This selection has the dual benefit for removing the human-in-the-loop element of manual curation of data, and significantly reducing the amount of data used to train the model. This tile selection also allows Theia to scale to arbitrarily large images.

Theia uses interaction terms as additional predictors in the model. An interaction term is the pixel-wise geometric mean of the target channel with a neighboring channel that contributes bleed-through. As a pre-processing step, pixel intensities are normalized to the $[0, 1]$ range, with low values representing background regions and high values representing foreground objects. Thus, interaction terms capture colocalized objects (objects that have been tagged with multiple fluorophores) and bleed-through of the target channel into neighboring channels. This leaves the primary terms to estimate bleed-through. While Theia uses both the primary and interaction terms to train the model, it only subtracts the primary terms to produce the corrected images. This leaves colocalized objects relatively untouched in the target channel. The loss for this GLM is:

$$\sum_i \left(y_i - \sum_j C_{ij}x_{ij} - \sum_j I_{ij}\sqrt{x_{ij}y_i}\right)^2 + \lambda \sum_{i,j} \left(|C_{ij}| + |I_{ij}|\right) \quad (3)$$

where the Cs are the coefficients for the primary terms and the Is are the coefficients for the interaction terms.

Equation 3 is adapted to the form in Equation 1 by dividing the first term by the number of pixels and the second term by the number of kernel coefficients. This re-scales the loss values to a range which is more manageable for neural network frameworks, like TensorFlow, to optimize.

As an additional optimization, it is important to note that not all channels need to be included as terms in the model. The tails of the emission spectra of fluorophores tend to quickly decrease from the peak emission wavelength, as seen in Figure 1A. Thus, light from a majority of fluorophores will bleed into only a small number of neighboring emission filters. If the "wavelength order" of emission spectra is known, Theia can take advantage of this information by ordering the channels by wavelength and then only fitting models with terms for a small number of adjacent channels rather than among all pairs of channels.

Datasets

The mechanism of bleed-through inspired the following approach to synthetic data generation, which resulted in a more realistic synthetic bleed-through data set for benchmarking. Complementing this dataset, metrics were developed to validate and compare Theia against competing approaches. Various combinations of parameters were used to generate synthetic data for controlled tests of Theia and to explore the various aspects of its performance. In addition, a real-world dataset was obtained from [12], henceforth referred to as the "RatBrain Data", and used for benchmarking all algorithms.

Synthetic Data

Synthetic data were generated by simulating various aspects of the process of capturing an image on a microscope, including small spatial aberrations, multiple fluorophores, corresponding filter-cubes, and simulated bleed-through. Spatial aberrations were generated using random mixing kernels of shape 3x3 with values sampled from a 2d Gaussian distribution whose peak is offset from the center of the kernel by a half pixel unit in a random direction. Each channel in an image is convolved with a new random spatial offset kernel. Then, bleed-through is computed using the sum of the target channel and the neighboring channels, weighted by the strength of the emission of the corresponding fluorophore through its filter cube. The resulting signals contain synthetic bleed-through from the adjacent channels.

Representative images of synthetic bleed-through are shown in Figure 3. These are limited to two-channel images for the purpose of demonstration. Five sets of synthetic data were generated. The first dataset consists of images containing only Poisson noise and is shown in Figure 3A. Theia is expected to perform poorly on this dataset since there is no distinction between bleed-through and noise.

The second and third datasets were generated using images from the TissueNet data [9]. For the second set, nuclear masks were randomly partitioned into two sets with ten percent being in common among the two sets. The overlap between the two sets represents cases where, in addition to one channel bleeding into another, there is also co-localization (i.e. objects stained with multiple fluorophores). These objects represent signal that should not be removed. This represents an ideal case where the background is always dim and the objects of interest all present with strong, uniform signal. This is shown in Figure 3B. The third set is the same as the second set except that the corresponding intensity images from the TissueNet data was used instead of the nuclear masks. This represents the case where the background is dim and the objects of interest are present with signals of varied strengths. This is shown in Figure 3C.

The fourth and fifth datasets were generated using the RatBrain dataset. The fourth dataset contains the same Field of View (FOV) from different slices in the Rat Brain data, and the strength of the original signal is left un-normalized.

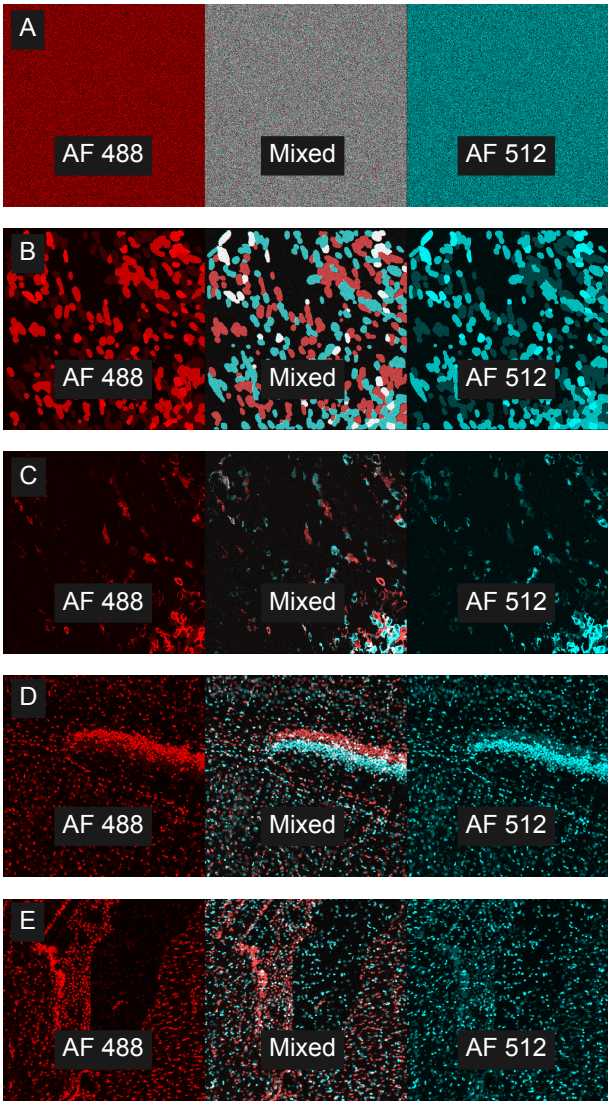

Figure 3. **Five datasets with synthetic bleed-through.** Row A shows images that only contain random Poisson noise. Row B shows images in which the objects have a bright and uniform signal. Row C shows the same images as row B but the objects have varied signal. Row D shows images from the same FOV from the Rat Brain data but with a large spatial offset. Row E shows images from different FOVs and channels from the Rat Brain data with the two images having been normalized for relative signal strength. The first and third columns show signal from Alexa Fluor 488 (red) and Alexa Fluor 512 (cyan) respectively, by simulating their corresponding full-width-half-max filters. The middle column shows the images form the first and third columns overlaid with each other so that the objects bleeding into other channels show in white, or with a significant tint, instead of red or cyan.

This represents the case where there is some background noise, the objects present with varied signal, and the different channels have different intensity profiles, i.e. the emission spectra of the fluorophores have their peaks at different heights. This is shown in Figure 3D. The fifth dataset contains different FOVs from the same replicate, and has had the signals normalized so that all channels have the same relative strength. This is shown in Figure 3E.

RatBrain Data

This dataset contains images of five slices from a rats brain, each of which was stained with ten different fluorophores. Starting with the FOVs, we applied the BaSiC algorithm [16] to correct for flat-field effects. Without this step, the centers of the FOVs would be brighter than the edges, significantly impacting the estimated bleed-through components. Next, we used the MIST image stitching tool [3, 5] to produce $41,906 \times 29,286$ sized images with 10 relevant channels for each of the five slices, forming the full RatBrain data.

We then applied Theia to estimate the bleed-through components for each channel and we subtracted away this component to correct for bleed-through. Since the authors provide information on the different fluorophores used [12] in the imaging process, we used this knowledge to order the channels by the wavelength of the emission filter of the corresponding fluorophores. This allowed us to restrict Theia to look for bleed-through contributions from one adjacent channel on each side on the wavelength scale.

4. Results

By its design, Theia is expected to perform well on diverse datasets. Two metrics were used to quantify the bleed-through correction in the images: Root Mean Squared Error (RMSE) and Pearson's Correlation Coefficient (PCC). Each metric was calculated on every pair of channels in the images before and after bleed-through correction. These metrics were chosen to measure relative similarity and the covariance among pairs of channels in the image before and after applying Theia. Our expectation is that when bleed-through is present, the signal across channels will be correlated and after bleed-through correction the correlation should decrease while the RMSE should increase (assuming co-staining is not perfect between channels).

Visually, Theia successfully removes synthetic bleed-through as expected (Figure 4A). Inspection of the learned filters confirms that the primary filters (bleed-through contributions) are well learned since channels with bleed-through components have parameters with larger magnitudes than channels that do not have bleed-through components (Figure 4B). Finally, quantitative analysis of RMSE and PCC shows expected changes before and after bleed-through correction (Figure 4C). Before bleed-through correction, RMSE is high and PCC is moderate. After bleed-through correction, RMSE is still relatively high but PCC is

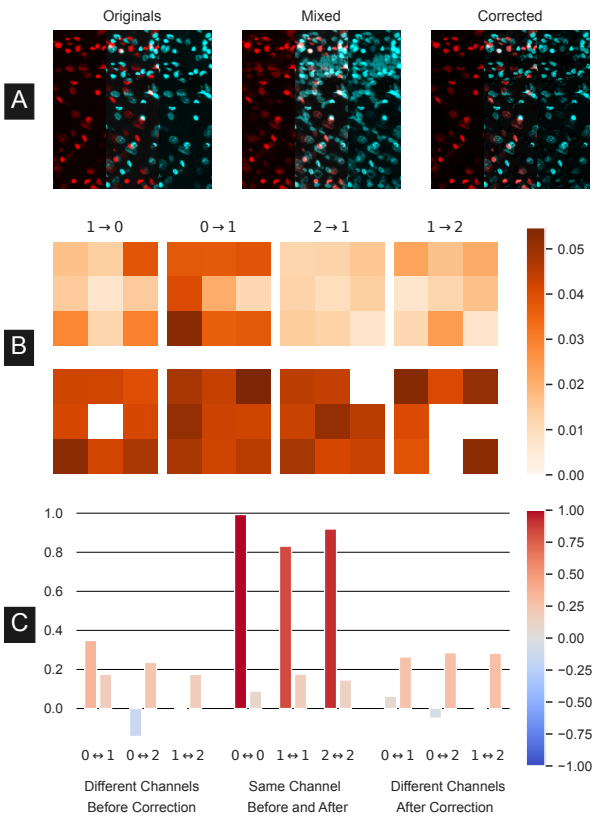

Figure 4. **Theia bleed-through correction.** A shows three representative original images, mixed images (i.e. after adding synthetic bleed-through) and corrected images (i.e. after applying Theia to the mixed images.) B shows the bleed-through correction kernels learned by Theia. The top row in B shows the primary (bleed-through) kernels and the bottom row shows the interaction kernels, while column $i \rightarrow j$ shows the kernels for bleed-through from channel i into channel j. C shows a grouped bar plot of bleed-through metrics on the images, where the pair of bars for each x-axis label show PCC on the left and RMSE on the right. The three groups, from left to right, are of metrics between *different* channels *before* bleed-through correction, between the *same* channels compared *before and after* bleed-through correction, and between *different* channels *after* bleed-through correction.

much lower across channels.

Figures 5 and 6 show a summary of an exhaustive and quantitative performance analysis of Theia on various datasets. For each dataset, Theia was used to generate corrected images, and this process was timed for 100 replicates on each dataset. The run-time for Theia scales linearly with increasing channel overlap and quadratically with increasing kernel size and image size. The run-time is independent of the combination of wavelengths and to image content between variants of the same dataset.

The *Poisson Noise* dataset is composed entirely of poisson-random noise in images, and we expect the bleed-through correction algorithm to be unable to identify bleed-through in the images. Indeed, Figure 6 shows that the PCC is nearly zero and RMSE is relatively high compared to the other four datasets. For the other four datasets, we consistently see that the corrected images have a high PCC with input images and also have a low RMSE.

Table 1. **Comparison of Theia to two methods using PCC and RMSE.** We compare Theia against LUMoS [13] and the version of LASSO from [17]. For PCC, high positive values are better whereas for RMSE, low values are better. The best performer for each metric in each column is in bold.

PCC	Theia	LUMoS	LASSO
Poisson	0.06	0.73	**0.76**
Uniform TissueNet	**0.91**	0.88	0.90
Varied TissueNet	**0.97**	0.92	0.80
Normalized RatBrain	**0.99**	0.95	0.82
RatBrain	**1.00**	0.96	0.92
RMSE	**Theia**	**LUMoS**	**LASSO**
Poisson	**0.23**	0.70	0.56
Uniform TissueNet	0.12	**0.10**	0.17
Varied TissueNet	**0.10**	0.24	0.22
Normalized RatBrain	0.15	**0.12**	0.29
RatBrain	**0.11**	0.15	0.19

Comparison with Other Methods

Table 1 shows a quantitative assessment of Theia against two other methods. The ImageJ plugin for LUMoS [13] was used for each of the five datasets. As a second point of comparison, the bleed-through correction portion of the workflow from [17] (which uses the LASSO model from scikit-learn) was used. With the exception of the dataset consisting entirely of Poisson noise, Theia outperforms these methods on all datasets as measured by PCC. When measured with RMSE, the results are split evenly between Theia and LUMoS. Unlike Theia, however, LUMoS assumes that fluorophores are spatially separated in images.

Comparison on the RatBrain Data

On an Arch Linux machine (kernel 6.1.3-arch1-1) with 16 cores (i7-211700KF) @ 4.9GHz, an NVIDIA GeForce RTX 3080, and 128G of memory, Theia took approximately 437 seconds on the Rat Brain Dataset. On a MacBookPro with the M1 Max and 64G of memory, Theia took approximately 516 seconds on the same data. On a Dell XPS 13 with 8 cores (i7-1065G7) @ 3.9GHz, 16G of memory, and no dedicated GPU, Theia took approximately 834 seconds on the same data. These benchmarks show that Theia does not require expensive hardware to run on large scale data. In comparison, the original pipeline [17] for this dataset took 6 hours on a machine with 32 cores, an NVIDIA GeForce

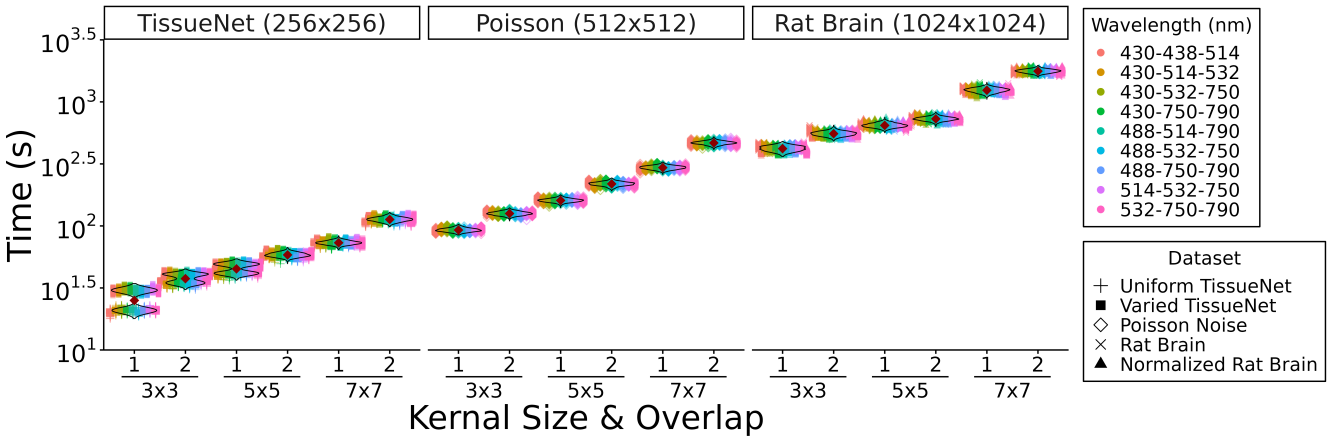

Figure 5. **Run-time benchmarks for Theia**, showing the combined time taken (in seconds) for learning and removing bleed-through with different hyper-parameters. Every combination of kernel size (3×3, 5×5 and 7×7) and channel overlap (1 and 2) is shown. Five different datasets and 9 different emission filter combos were used, covering a wide array of data structure and bleed-through amount. The five datasets are grouped into three facets. 100 runs were performed for each combination of parameters. As shown in the graphs, run-time is mostly independent of wavelength combination and does not vary within the two variants of the same dataset. Run-time does increase with increases in channel overlap, kernel size and image size.

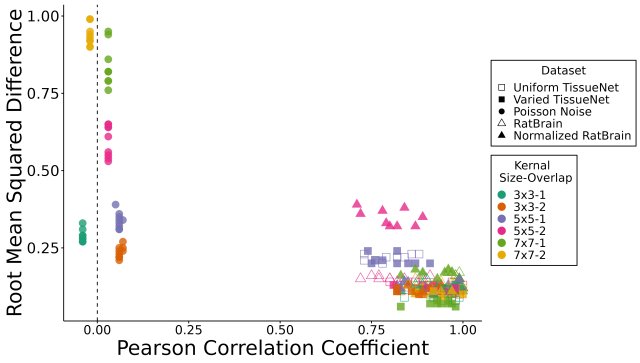

Figure 6. **Theia bleed-through metrics.** The y-axis shows the mean RMSE while the x-axis shows the mean PCC between channels from each image before and after bleed-through correction. Random Poisson noise exhibits low PCC and high RMSE values. For each of the other 4 datasets, the PCC is very high while RMSE is low, indicating effective removal of bleed-through without loss of important information from each channel. Each data point is the mean of 100 runs. The standard deviation among the 100 measurements was always smaller than a factor of 10^{-3} of the mean and is not shown on this plot.

GTX 1080 Ti, and 256G of memory, and required manual RoI annotations. Theia requires less manual intervention (no RoI selection) and has over an order of magnitude better speed (26x) on the same dataset, despite running on far smaller computational resources.

5. Discussion

Theia is a bleed-through correction algorithm inspired by LASSO and Siamese Networks. Theia has a novel Dynamically Branched Neural Network architecture with a LASSO regression loss objective for use in multi-channel images. Using this architecture Theia calculates kernels that invert spectral mixing, rather than typical neural network outputs such as images or embeddings.

Theia addresses many of the drawbacks of other algorithms for bleed-through correction [11–13, 22]. It makes no assumptions about the spacial locality, or overlap, of objects tagged with different fluorophores, removes the need for manual RoI selection, and does not require any reference images or the emission spectra of fluorophores. The only requirement is that there be a one-to-one correspondence between the channels and the fluorophores. An optional requirement, for an optimization step, is that one know the relative wavelength ordering of the channels.

Future work may alleviate the latter requirement, for example, through the use of different (higher-order) interaction components.

Additional work may be conducted to use the learned filters for signal purification. While the bleed-through component is "noise" to be removed from other channels, it may be useful signal for the channel producing the bleed-through effect.

Further future work can explore the effects of having a higher fraction of colocalized objects in different channels.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 4

[2] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006. 3

[3] Timothy Blattner, Walid Keyrouz, Joe Chalfoun, Bertrand Stivalet, Mary Brady, and Shujia Zhou. A hybrid cpu-gpu system for stitching large scale optical microscopy images. In *2014 43rd International Conference on Parallel Processing*, pages 1–9. IEEE, 2014. 6

[4] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6, 1993. 3

[5] Joe Chalfoun, Michael Majurski, Tim Blattner, Kiran Bhadriraju, Walid Keyrouz, Peter Bajcsy, and Mary Brady. Mist: accurate and scalable microscopy image stitching tool with stage modeling and error minimization. *Scientific reports*, 7(1):1–10, 2017. 6

[6] Davide Chicco. Siamese neural networks: An overview. *Artificial Neural Networks*, pages 73–94, 2021. 3

[7] François Chollet et al. Keras. https://keras.io, 2015. 4

[8] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005. 3

[9] Noah F Greenwald, Geneva Miller, Erick Moen, Alex Kong, Adam Kagel, Thomas Dougherty, Christine Camacho Fullaway, Brianna J McIntosh, Ke Xuan Leow, Morgan Sarah Schwartz, et al. Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. *Nature biotechnology*, 40(4):555–565, 2022. 5

[10] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000. 3

[11] Dahan Kim, Nikki M Curthoys, Matthew T Parent, and Samuel T Hess. Bleed-through correction for rendering and correlation analysis in multi-colour localization microscopy. *Journal of optics*, 15(9):094011, 2013. 2, 4, 8

[12] Dragan Maric, Jahandar Jahanipour, Xiaoyang Rebecca Li, Aditi Singh, Aryan Mobiny, Hien Van Nguyen, Andrea Sedlock, Kedar Grama, and Badrinath Roysam. Whole-brain tissue mapping toolkit using large-scale highly multiplexed immunofluorescence imaging and deep neural networks. *Nature communications*, 12(1):1–12, 2021. 2, 3, 4, 5, 6, 8

[13] Tristan D McRae, David Oleksyn, Jim Miller, and Yu-Rong Gao. Robust blind spectral unmixing for fluorescence microscopy using unsupervised learning. *Plos one*, 14(12):e0225410, 2019. 2, 3, 4, 7, 8

[14] HP D Paulo, José C Pereira, Nádia R Svizero, Frederick A Rueggeberg, and David H Pashley. Use of fluorescent compounds in assessing bonded resin-based restorations: a literature review. *Journal of dentistry*, 34(9):623–634, 2006. 1

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 4

[16] Tingying Peng, Kurt Thorn, Timm Schroeder, Lichao Wang, Fabian J Theis, Carsten Marr, and Nassir Navab. A basic tool for background and shading correction of optical microscopy images. *Nature communications*, 8(1):1–7, 2017. 6

[17] RoysamLab. Whole brain analysis pipeline. 2020. 4, 7

[18] Reshma Shakya, Tam Hong Nguyen, Nigel Waterhouse, and Rajiv Khanna. Immune contexture analysis in immunooncology: applications and challenges of multiplex fluorescent immunohistochemistry. *Clinical & Translational Immunology*, 9(10):e1183, 2020. 2

[19] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014. 3

[20] Fabian J Theis, Richard Neher, and Andre Zeug. Blind decomposition of spectral imaging microscopy: A study on artificial and real test data. In *Independent Component Analysis and Signal Separation: 8th International Conference, ICA 2009, Paraty, Brazil, March 15-18, 2009. Proceedings 8*, pages 548–556. Springer, 2009. 3

[21] Robert Tibshirani. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(3):273–282, 2011. 3

[22] Alex M Valm, Rudolf Oldenbourg, and Gary G Borisy. Multiplexed spectral imaging of 120 different fluorescent labels. *PloS one*, 11(7):e0158495, 2016. 2, 3, 4, 8

[23] Jennifer C Waters. Accuracy and precision in quantitative fluorescence microscopy, 2009. 1

[24] Timo Zimmermann, Joanne Marrison, Karen Hogg, and Peter O'Toole. Clearing up the signal: spectral imaging and linear unmixing in fluorescence microscopy. In *Confocal microscopy*, pages 129–148. Springer, 2014. 1