Dawnn: single-cell differential abundance with neural networks

George T. Hall[1,*] and Sergi Castellano[1,2]

[1]Genetics and Genomic Medicine, UCL Great Ormond Street Institute of Child Health, Zayed Centre for Research into Rare Disease in Children, 20 Guilford Street, London, WC1N 1DZ, United Kingdom

[2]UCL Genomics, Zayed Centre for Research into Rare Disease in Children, 20 Guilford Street, London, WC1N 1DZ, United Kingdom

[*]Corresponding author: `george.hall@ucl.ac.uk`

Abstract

Analysis of single-cell transcriptomes can identify cell populations more abundant in one sample or condition than another. However, existing methods to discover them suffer from either low discovery rates or high rates of false positives. We introduce Dawnn, a deep neural network able to find differential abundance with higher accuracy than current tools, both on simulated and biological datasets. Further, we demonstrate that Dawnn recovers published findings, promising novel biological insights at single-cell resolution.

1 Introduction

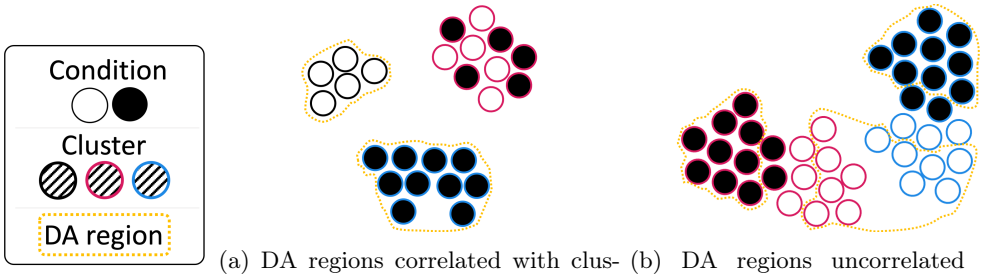
Single-cell RNA sequencing allows the comparison of samples using the transcriptomes of individual cells. A common task is to identify cell populations that exhibit *differential abundance* (DA) across samples or conditions in a study, for example enriched or depleted cell populations between treated and control conditions in a drug trial, or differences between patients under the same treatment or treatment time points.

Despite its importance, DA analysis remains a challenge. A naïve approach to identifying regions exhibiting DA is to create clusters and identify those dominated by just one cell population, sample, or condition. However, this method can perform poorly if DA is split across multiple clusters, since the signal of enrichment or depletion is split and therefore becomes challenging to detect (Figure 1).

To address this problem, recent methods such as Milo [1], DA-seq [2], and MELD [3] attempt to identify DA without performing clustering whilst moderating the false discovery rate (FDR) [1].

In this paper, we present *Dawnn* (*differential abundance with neural networks*), which improves DA identification over existing methods whilst matching their low FDR. Dawnn uses a deep neural network model that has been trained to estimate the relative abundance of cells from each sample or condition in a cell's neighbourhood. Our approach differs from existing ones since, instead of fitting traditional statistical models to a nearest-neighbour graph of cell labels, we have instead trained our algorithm to predict the probability with which each cell was drawn from a given sample or condition using simulated training datasets.

We evaluate Dawnn's performance using the benchmarking procedures and datasets introduced by Dann *et al.* to assess Milo [1]. These comprise simulated and biological datasets of varying complexity. We also evaluate our approach on three additional published real-world datasets. We find that Dawnn yields higher true positive rates than existing methods whilst maintaining low rates of false discovery (the proportion of cells incorrectly claimed to be in populations with differential abundance). We also



(a) DA regions correlated with clusters.

(b) DA regions uncorrelated with clusters.

Figure 1: Successful DA identification using clustering requires DA regions to correlate with assigned clusters. The fill and outline colours of a cell indicate condition and cluster, respectively. Regions of DA are outlined in yellow (in this figure, a cell is defined as in a DA region if it and all immediate neighbours are from the same sample). **a** If regions of DA correlate with clusters, then they can be detected using clustering. **b** If regions of DA do not correlate with clusters, then they may not be detected using clustering.

demonstrate that, as shown by Dann *et al.* for Milo, Dawnn is capable of recovering and extending published findings of differential abundance [4], implying that its good benchmarking performance generalises to real data.

In conclusion, Dawnn better identifies DA between cell populations, samples or conditions than existing state-of-the-art algorithms and has a similarly low FDR. It can thus lead to novel biological insights by highlighting which populations are perturbed and thus worthy of further analysis. Dawnn is available as an R package[1].

2 Results

2.1 Dawnn detects differential abundance using neural networks

Dawnn is designed to identify regions of differential abundance in cell populations comprising two samples or conditions. It differs from existing algorithms in its methodology but employs the same basic assumptions, namely that each cell is assigned one of two labels (S_a or S_b) and is associated with an individual unknown probability p of having the label S_a (in effect, of being drawn from the sample or condition associated with label S_a). We denote as $\vec{p}$ the vector of all probabilities p associated with a dataset. A given single-cell transcriptomic dataset is thus an instantiation of labellings generated according to $\vec{p}$. Under this model, the task of detecting DA regions is equivalent to that of identifying cells for which p differs substantially from the null distribution.

In general, differential abundance detection methods estimate $\vec{p}$ from a distribution of sample labels on a K-nearest neighbour graph. Batch effects can be mitigated by generating the KNN from a batch-corrected dimensionality reduction (e.g. Harmony [5]). Existing DA detection algorithms estimate $\vec{p}$ by fitting a model (e.g. Milo's generalised linear or DA-seq's logistic regression models) to this distribution. Dawnn, on the other hand, is a neural network (Section 6.1) that has been taught to estimate $\vec{p}$ using simulated training data. Dawnn first estimates $\vec{p}$ from the sample labels of the 1,000 nearest neighbours of each cell, then generates a null distribution of labels to represent a dataset with no differential abundance by reassigning labels from a distribution where both are equally prevalent. It then estimates $\vec{p}$ for this null distribution

[1] `https://github.com/george-hall-ucl/dawnn`

dataset. For each cell, Dawnn tests the null hypothesis that it is in a region with no differential abundance, calculating a p-value by fitting a beta distribution to the estimates of $\vec{p}$ from the null distribution data and calculating the probability of observing at least such an extreme a p under this distribution. Dawnn controls the false discovery rate (FDR), the proportion of cells incorrectly classified as belonging to regions exhibiting DA, using the Benjamini–Yekutieli procedure [6], a variant of the Benjamini-Hochberg procedure [7] that does not assume independence between hypotheses. This more conservative FDR-controlling mechanism is necessary since the hypotheses associated with neighbouring cells will be correlated. Pseudocode for Dawnn is given in Algorithm 1.

Algorithm 1: Dawnn (*differential abundance with neural networks*)

Input Vector ℓ containing each cell's associated label
Input Matrix M containing labels from ℓ for 1,000 nearest neighbours of each cell
Input Neural network model `NN` trained to estimate $\vec{p}$ from M

```
q⃗ ← NN (M) // use NN to estimate p⃗ from M
ℓ′ ← uniform distribution of the two unique labels in ℓ
M′ ← matrix of labels from ℓ′ for 1,000 nearest neighbours of each cell
q⃗′ ← NN (M′) // use NN to estimate p⃗ from M′
(α′, β′) ← parameters of beta distribution fitted to q⃗′
ρ ← p-values of observed p⃗ according to Beta (α′, β′)
Determine DA cells using ρ and Benjamini–Yekutieli procedure
```

We selected the architecture of Dawnn's neural network from a pool of candidates assessed using K-fold cross validation (Section 6.2). Our algorithm training and selection procedures were designed to yield an unbiased model capable of estimating $\vec{p}$ across many real-world datasets. To maximise generalisation and avoid bias during benchmarking, we designed Dawnn's training set to make no assumptions about the overall structure of cell populations and only weak assumptions about the relationship between the labels of cells within each neighbourhood (Section 4.1.1).

2.2 Dawnn detects differentially abundant regions with higher accuracy than existing methods

We benchmarked Dawnn against Milo and DA-seq, since these were the two best-performing approaches in Dann *et al.* [1]. We follow their benchmarking methodology (Section 4.2.2), creating simulated and real datasets using their described approach [1]. We further assessed real-world performance using three additional biological datasets: one from a publication of ours examining the effects of accelerating skin sheet growth [8]; another from a publication characterising organoids of bile ducts [9]; and a large dataset comprising cells from the heart in a study of congenital heart disease [10]. Table 1 summarises our benchmarking datasets.

The simulated and experimental datasets provide the structure of cell populations to which simulated labels can be assigned, providing the ground truth necessary to measure algorithm performance.

For each simulated dataset, six cell populations were created. For each population, multiple distributions of sample labels were simulated. Each label distribution was defined by the subpopulation exhibiting differential abundance and the extent of differential abundance in this subpopulation. For each choice of differentially abundant subpopulation, three sets of sample labels were simulated for each maximum $\log_2$-fold change in the set $\{1.22, 1.58, 2, 2.5, 3.17, 4.25\}$. These are the same $\log_2$-fold changes employed by Dann *et al.*

A similar methodology was used to simulate sample labels for datasets from real experiments. For these, eight subpopulations were chosen to be differentially abundant.

Dataset	Source	Number of cells	Number of cell types
Discrete clusters	Simulation [1]	2,700	NA
Linear trajectory	Simulation [1]	10,000	NA
Branching trajectory	Simulation [1]	7,500	NA
Mouse gastrulation	Experiment [11]	64,018	37
Accelerated keratinocytes	Experiment [8]	18,787	3
Bile duct organoids	Experiment [9]	72,803	1
Congenital heart disease	Experiment [10]	157,293	14

Table 1: Test datasets used to assess true positive and false discovery rates.

These subpopulations corresponded to annotated cell types in the mouse gastrulation dataset and to clusters in the keratinocyte, organoid, and heart datasets. The maximum log-fold changes were identical to those in the simulated datasets.

We assessed algorithm performance by calculating the true positive rate (TPR), the proportion of cells correctly identified as being in a DA region, and false discovery rate (FDR), the proportion of cells incorrectly classified as belonging to a DA region. Dawnn and Milo allow a maximum acceptable FDR to be specified. We set this threshold to 10%, as in Dann *et al.*

2.2.1 Simulated discrete clusters

In the simulated discrete clusters dataset, Dawnn achieved higher TPRs than Milo and DA-seq (Figure 2a). The FDRs of Dawnn were comparable to those of DA-seq (in most cases being almost indistinguishable from 0), whilst Milo yielded higher FDRs (often above the desired 10% maximum). Both DA-seq and Milo did not identify a high proportion of DA regions when the simulated fold change in these regions was small. Dawnn, on the other hand, exhibited good performance for all simulated fold changes.

2.2.2 Simulated linear trajectory

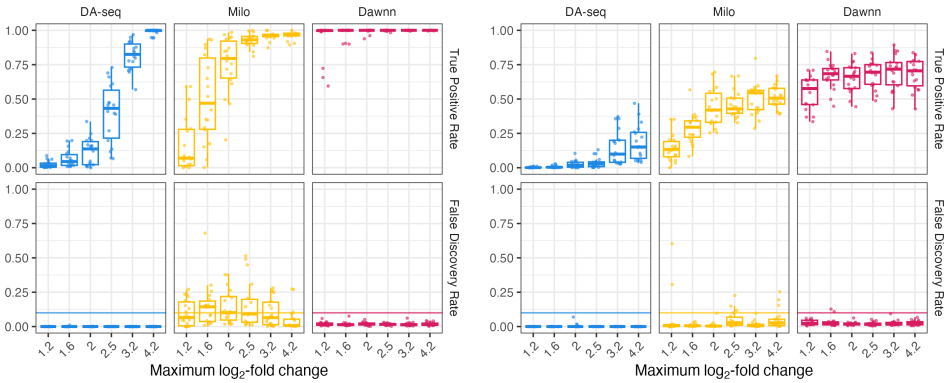
On the simulated linear trajectory dataset, Dawnn once more yielded higher TPRs than those from DA-seq and Milo whilst maintaining FDRs close to 0 (Figure 2b). However, the TPRs of all three algorithms were substantially lower than in the simulated discrete cluster case, indicating a more complex scenario.

2.2.3 Simulated branching trajectory

For the simulated branching trajectory dataset, Dawnn again exhibited higher TPRs than Milo and DA-seq, albeit with larger variation than for the other two simulated datasets (Figure 2c). Dawnn again maintained an FDR predominantly below 10%, unlike Milo, which often exceeded this threshold.

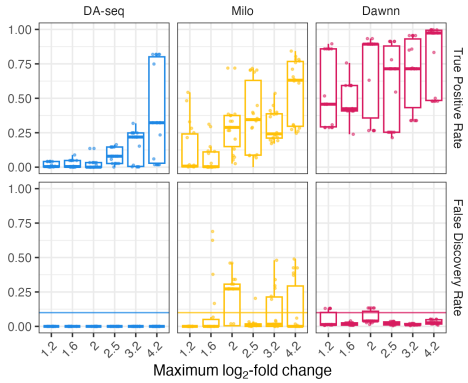
2.2.4 Mouse gastrulation

On the mouse gastrulation dataset [11], Dawnn exhibited considerably higher TPRs than both Milo and DA-seq, incurring only a marginal increase in FDR (Figure 3a). On average, Dawnn's FDR remained below the 10% threshold. Whilst Dawnn's FDR was generally slightly higher than that of Milo, it exceeded 10% less often. DA-seq exhibited the lowest FDR.



(a) Discrete clusters

(b) Linear trajectory



(c) Branching trajectory

Figure 2: True positive and false discovery rates of DA-seq, Milo, and Dawnn on simulated datasets. Dawnn consistently identifies a larger proportion of DA cell populations whilst maintaining a FDR below the 10% threshold.

2.2.5 Growth-accelerated keratinocytes

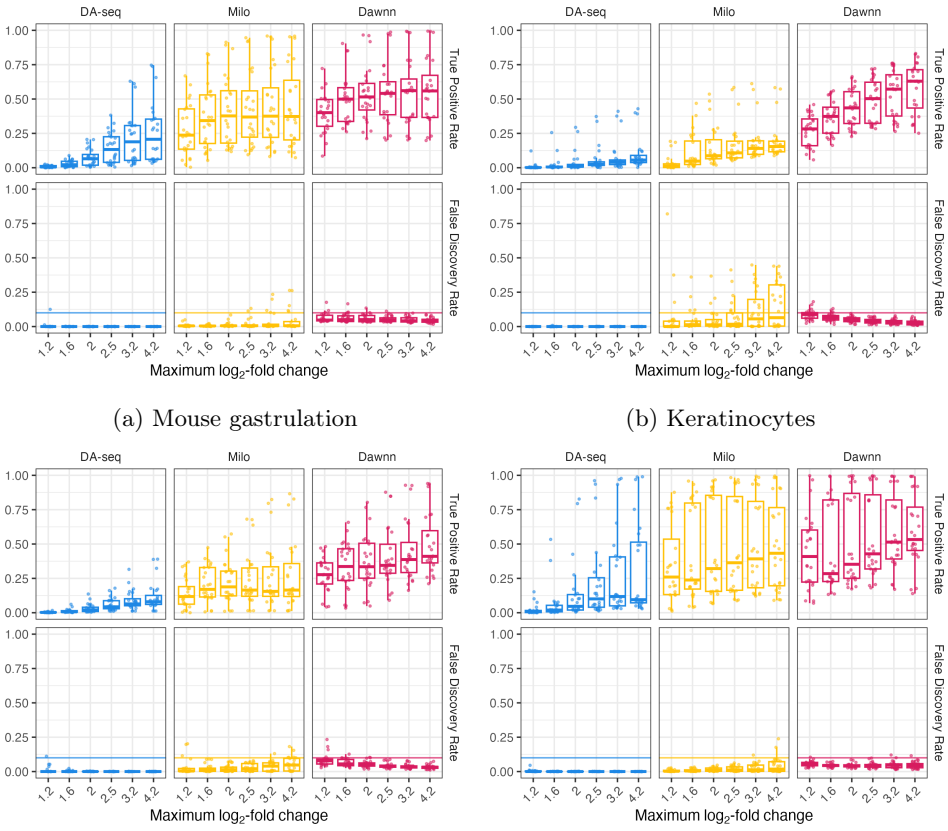
On the keratinocyte dataset [8], we again found that Dawnn identified a larger proportion of DA regions than Milo and DA-seq (Figure 3b). Dawnn maintained a far lower FDR than Milo, which was often over the 10% desired threshold. DA-seq again had the lowest FDR.

2.2.6 Bile duct organoids

On the dataset derived from organoids of bile ducts [9], Dawnn identified a higher proportion of DA populations than Milo and DA-seq (Figure 3c). It once again maintained an FDR below the specified 10%. The lowest FDR was that of DA-seq.

2.2.7 Heart cells

Finally, in the single-nucleus heart cell dataset, we again observed that Dawnn exhibited a higher TPR than Milo and DA-seq. However, the TPRs of each method were more variable than for smaller test datasets (Figure 3d). All three methods maintained FDRs predominantly below 10%.



(a) Mouse gastrulation

(b) Keratinocytes

(c) Bile duct organoids

(d) Heart

Figure 3: True positive and false discovery rates of DA-seq, Milo, and Dawnn on real biological datasets. As with the simulated cell population structures, for the real-world data, Dawnn is capable of identifying higher proportions of differential abundance than Milo and DA-seq whilst maintaining a low FDR (below 10%).

2.2.8 Results summary

Overall, we found that Dawnn was able to identify a larger proportion of DA regions than both Milo and DA-seq. With higher TPRs than existing methods, Dawnn was still able to maintain low FDRs (generally below the specified 10% threshold). DA-seq maintained FDRs close to 0 throughout, but suffered from TPRs generally far lower than those in the other two methods. The TPRs of Milo were generally between those of DA-seq and Dawnn, however it sometimes suffered from FDRs exceeding the 10% threshold. We validated these conclusions with statistical analyses (Section 5 in supplementary materials)

2.3 Dawnn estimates the local distribution of fold changes more accurately than existing methods

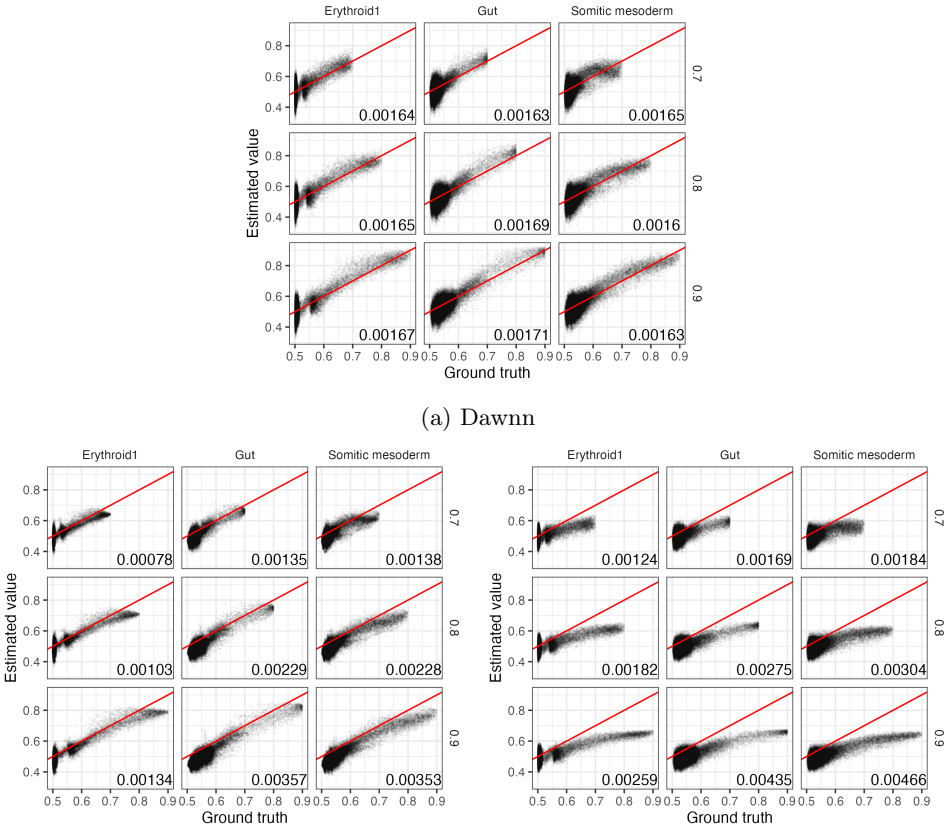
Since MELD cannot automatically classify cells as DA, we followed Dann *et al.* and benchmarked against it by comparing the estimated $\vec{p}$ vectors against the ground truth (i.e. the $\vec{p}$ used to simulate the sample labels). Figure 4 shows this relationship for nine simulations of sample labels on the mouse gastrulation dataset. In the case of an algorithm with perfect performance, all points would be on the red line indicating $y = x$ (i.e. for each cell, the predicted p matches the ground truth). Each plot also shows the mean squared error (MSE), the mean squared difference between each estimated p and its associated ground truth.

For all test instances, the MSE of Dawnn's $\vec{p}$ estimates is at most 0.00171 (i.e. on average Dawnn's estimated $\vec{p}$ is within 0.042 of the ground truth). On the other hand, the maximum MSEs of MELD and DA-seq are 0.00357 and 0.00466, respectively. The MSEs of Dawnn's estimates are less variable than those of MELD and DA-seq. MELD has the smallest MSE for five of the benchmarking instances, whilst Dawnn has the smallest on the remaining four. Dawnn has a smaller MSE than DA-seq in all but one test instance. MELD provides more accurate estimates of $\vec{p}$ in the best case, whilst Dawnn has the best performance in the worst case. In particular, MELD and DA-seq underestimate $\vec{p}$ for cells in regions with high DA (i.e. with high p), whereas Dawnn overestimates $\vec{p}$ for these cells.

2.4 Dawnn recovers published claims of differential abundance

Dann *et al.* demonstrate that Milo is capable of recovering findings from Ramachandran *et al.*, who compared samples of healthy and cirrhotic livers [4]. We found that both Dawnn and DA-seq are able to recover these findings at the single-cell level rather than the neighbourhoods operated on by Milo. Figure 5 compares the outputs of these three methods for this dataset.

For cell types labelled as differentially abundant by Ramachandran *et al.*, the outputs of all three methods report largely similar extents of differential abundance. Within these groups of cells, as Dann *et al.* reported for Milo, both Dawnn and DA-seq identify sub-populations differentially abundant in the opposite direction to the larger population. In some cases, the cell-level nature of Dawnn and DA-seq allows them to identify populations that are missed by Milo, such as T Cells (5) and PDCs. For all cell types, Dawnn labels more cells as belonging to regions exhibiting differential abundance than other approaches, using a log-fold change cutoff of approximately ± 0.5 rather than the approximately ± 1.5 and ± 2 in DA-seq and Milo, respectively.



(a) Dawnn

(b) MELD

(c) DA-seq

Figure 4: Estimates of $\vec{p}$ vs. ground truth for Dawnn (a), MELD (b), and DA-seq (c). Within each plot, rows correspond to different maximum upregulation in the differentially abundant subpopulation (i.e. $\max \vec{p}$) and columns to different cell types chosen to exhibit differential abundance. Mean squared errors are shown in the bottom right of each plot.

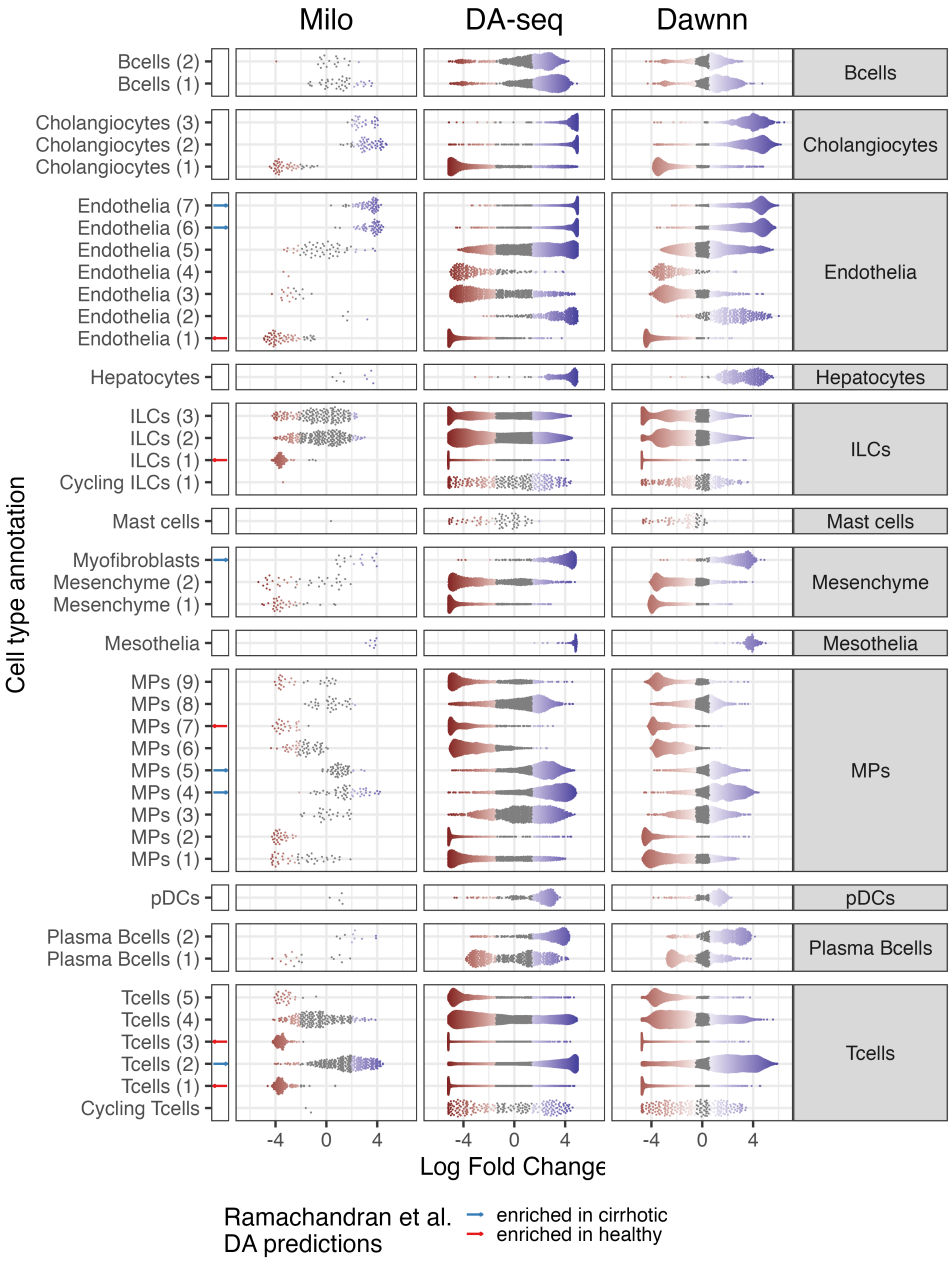


Figure 5: Log-fold changes estimated by Milo (L), DA-seq (C), and Dawnn (R) on the liver cirrhosis dataset of Ramachandran *et al.* [4].

2.5 Dawnn and DA-seq have similar runtimes but are slower than Milo

We compared the runtime of Dawnn against that of DA-seq and Milo on different sizes of datasets (Figure 9). Importantly, for datasets up to 60,000 cells, the runtime of all methods was below 16 minutes on a standard laptop, making them practical for most studies. We found the fastest method to be Milo (which tests for DA within neighbourhoods rather than at the single-cell level), with a runtime approximately 50% smaller than that of DA-seq and Dawnn (which both test for DA at the single-cell level). Dawnn was approximately 10% slower than DA-seq. Crucially, the runtime of Dawnn scales linearly with the number of cells in the dataset, allowing it to be used on large datasets without becoming prohibitively slow.

3 Discussion

In single-cell transcriptomics, the identification of cell populations more abundant in one sample or condition than another is a common and important task. However, current methods are unable to do so with high rates of discovery whilst avoiding false positives. Here we present Dawnn, an algorithm that identifies a higher proportion of differential abundance than existing state-of-the-art methods whilst maintaining a low rate of false discovery.

One explanation for why Dawnn achieves higher rates of discovery than current approaches is due to how it determines statistical significance. Whilst DA-seq, for example, only classifies as DA those cells with a more extreme estimate of p than any of those in the dataset with shuffled labels, Dawnn calculates the probability of observing at least such an extreme score with respect to a null distribution of shuffled sample labels generated according to a uniform distribution. When designing Dawnn, we found that applying the approach used by DA-seq made Dawnn unnecessarily conservative: far fewer cells were classified as DA (i.e. the TPR was decreased for a only minimal decrease in FDR). This finding suggests that employing Dawnn's approach in other methods may improve their performance.

Dawnn's model inputs (sample labels rather than the label proportions used by DA-seq) may also have contributed to its good performance. This approach allows a flexible model, such as a neural network, to use the most useful data representation since it can internally convert to a proportion-based representation if that yields greater accuracy. Using sample labels as input loses no information, whereas using label proportions does (Section 7). We thus recommend that, in future DA detection algorithms, label-based representations be favoured over proportion-based ones.

Dawnn's improved performance over Milo might be explained by its cell-level resolution. Its more consistent control of the false discovery rate may be partially due to its use of the Benjamini–Yekutieli procedure, which does not assume independence between the extent of DA expressed by neighbouring cells.

Testing the ability of DA detection algorithms to recover published findings for a cirrhotic liver dataset (Section 2.4) reveals further advantages of Dawnn over existing methods. First, the threshold for the point at which the fold change between sample label proportions is statistically significant is considerably smaller in Dawnn than Milo and DA-seq, despite both Dawnn and Milo attempting to be as conservative as one another (i.e. to stay under an FDR of 10%). Therefore, Dawnn can identify more DA populations of interest whilst preserving a low false discovery rate (as demonstrated in Section 2.2). Second, the single-cell resolution of Dawnn and DA-seq allows them to detect more sub-populations exhibiting DA than is possible with the neighbourhood-resolution Milo, since neighbourhood-level averaging of fold change estimates leads to signal from small populations being lost.

We found in Section 2.5 that, unsurprisingly, DA detection methods operating at the single-cell level are slower than their neighbourhood-resolution counterparts due to

the larger number of data points on which they operate. Despite this, we believe that the ∼50% larger runtime is justified since it allows the single-cell resolution of the data to be exploited fully. Since it remains possible to run the single-cell-resolution methods on a matter of minutes on a standard laptop, we believe this slowdown to be irrelevant in practice.

The pre-trained nature of Dawnn comes at the expense of flexibility in terms of experimental design. Unlike Milo, Dawnn can only be used to assess differential abundance in studies involving two samples or conditions (a limitation shared by DA-seq). Further, Milo can control for batch effects by incorporating covariates related to the experimental design in its statistical model. Dawnn's pre-trained model again prevents this, although a batch-corrected dimensionality reduction could be used when constructing the KNN graph (in fact, one is used for the growth-accelerated keratinocyte dataset). Similarly, Dawnn's pre-trained model requires a nearest neighbour graph of 1,000 neighbours for each cell, making the choice of the K parameter hard-coded, unlike in other methods. We chose a large neighbourhood size as this allows differential abundance to be detected at a range of scales (as in DA-seq). Dawnn's performance may be improved by allowing this parameter to be optimised by the user for their dataset. Despite these decreases in flexibility in terms of experimental design and parameter choice, we believe that Dawnn's ability to identify DA with higher accuracy than existing methods means that it will prove invaluable in the typical assessment of two samples or experimental conditions. Further, since Dawnn's training code is open-source[2], its neural network can be retrained with $K \neq 1000$ if desired.

Whilst we have shown Dawnn to outperform existing methods following benchmarking procedures established by Dann *et al.*, there is a risk that its good performance arises simply because it has been trained to estimate $\vec{p}$ using a training dataset generated in a similar manner to the benchmarking datasets, thereby biasing the process in its favour. Clearly, an algorithm trained on a dataset similar to the one used for benchmarking may yield good performance as a result of biased training. For this reason, we removed as many assumptions as possible from the training set generation (Section 4.2.2), thus minimising bias towards the benchmarking one. Further, we assessed Dawnn's performance on a range representation of single-cell datasets (Table 1) to ensure that its good performance generalises to unseen data.

When designing Dawnn, we investigated whether one-dimensional convolutional neural networks or graph neural networks [12] could reduce the error when estimating $\vec{p}$ by exploiting local neighbourhood structure. However, we found that this was not the case, at least in our hands. One explanation for this is that this was caused by the lack of structure in the training data, meaning that performance could not be improved by attempting to exploit it.

When training neural networks, it is common to use a technique called dropout to improve generalisation of the learned model. This approach randomly disregards some elements of the model during training, which slows improvement but prevents overfitting to the training data. However, we found dropout to have a negative effect when training Dawnn (Table 2), possibly because there is little opportunity to overfit the training data due to its lack of structure. We therefore believe that the negative effect of dropout indicates, as with the poor performance of convolutional and graph neural networks, that we have succeeded in our aim to create training data with little structure and removed benchmarking bias in Dawnn's favour.

We are thus confident that Dawnn's strong benchmarking performance is not because its training dataset was similar to that used for assessing its performance, but rather because it performs well on real-world datasets, as shown on the liver cirrhosis dataset (Section 2.4). We believe that the benchmarking methodology devised by Dann *et al.* and used here captures the underlying structure that must be approximated by any DA detection algorithm, and therefore there is no alternative approach that maintains the power of this benchmarking methodology. By expanding the num-

[2] `https://doi.org/10.5522/04/22633606`

ber of biological datasets, we have further measured the real-world performance of these methods beyond the insights from Dann *et al.* We believe that this expanded benchmarking dataset[3] will allow other DA detection methods to be robustly assessed.

In conclusion, Dawnn provides increased resolution in the identification of differential abundance in single-cell experiments without additional false discoveries. This will allow a more fine-grained investigation of the subtle biology of individual cells and cell populations.

Supplementary information Supplementary materials are contained in the appendix. Resources to reproduce all results in this paper are available online[4].

Acknowledgements GTH and SC are funded by the NIHR Great Ormond Street Biomedical Research Centre. We would like to acknowledge that benchmarking in this work was largely based on that from Dann *et al.* [1], whose clear methodology and code we appreciated enormously.

References

[1] E. Dann, N. C. Henderson, S. A. Teichmann, M. D. Morgan, and J. C. Marioni, "Differential abundance testing on single-cell data using k-nearest neighbor graphs," *Nature Biotechnology*, vol. 40, no. 2, pp. 245–253, 2022.

[2] J. Zhao, A. Jaffe, H. Li, O. Lindenbaum, E. Sefik, R. Jackson, X. Cheng, R. A. Flavell, and Y. Kluger, "Detection of differentially abundant cell subpopulations in scrna-seq data," *Proceedings of the National Academy of Sciences*, vol. 118, no. 22, p. e2100293118, 2021.

[3] D. B. Burkhardt, J. S. Stanley, A. Tong, A. L. Perdigoto, S. A. Gigante, K. C. Herold, G. Wolf, A. J. Giraldez, D. van Dijk, and S. Krishnaswamy, "Quantifying the effect of experimental perturbations at single-cell resolution," *Nature Biotechnology*, vol. 39, no. 5, pp. 619–629, 2021.

[4] P. Ramachandran, R. Dobie, J. Wilson-Kanamori, E. Dora, B. Henderson, N. Luu, J. Portman, K. Matchett, M. Brice, J. Marwick, *et al.*, "Resolving the fibrotic niche of human liver cirrhosis at single-cell level," *Nature*, vol. 575, no. 7783, pp. 512–518, 2019.

[5] I. Korsunsky, N. Millard, J. Fan, K. Slowikowski, F. Zhang, K. Wei, Y. Baglaenko, M. Brenner, P. Loh, and S. Raychaudhuri, "Fast, sensitive and accurate integration of single-cell data with harmony," *Nature Methods*, vol. 16, no. 12, pp. 1289–1296, 2019.

[6] Y. Benjamini and D. Yekutieli, "The control of the false discovery rate in multiple testing under dependency," *Annals of Statistics*, pp. 1165–1188, 2001.

[7] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 57, no. 1, pp. 289–300, 1995.

[8] V. Jayarajan, G. T. Hall, T. Xenakis, N. Bulstrode, D. Moulding, S. Castellano, and W. Di, "Short-term treatment with rho-associated kinase inhibitor preserves keratinocyte stem cell characteristics in vitro," *Cells*, vol. 12, no. 3, p. 346, 2023.

[9] F. Sampaziotis, D. Muraro, O. C. Tysoe, S. Sawiak, T. E. Beach, E. M. Godfrey, S. S. Upponi, T. Brevini, B. T. Wesley, J. Garcia-Bernardo, *et al.*, "Cholangiocyte organoids can repair bile ducts after transplantation in the human liver," *Science*, vol. 371, no. 6531, pp. 839–846, 2021.

[3] `https://rdr.ucl.ac.uk/projects/Dawnn_manuscript_files/161707`
[4] `https://github.com/george-hall-ucl/dawnn_paper_resources`

[10] M. C. Hill, Z. A. Kadow, H. Long, Y. Morikawa, T. J. Martin, E. J. Birks, K. S. Campbell, J. Nerbonne, K. Lavine, L. Wadhwa, *et al.*, "Integrated multi-omic characterization of congenital heart disease," *Nature*, vol. 608, no. 7921, pp. 181–191, 2022.

[11] B. Pijuan-Sala, J. A. Griffiths, C. Guibentif, T. W. Hiscock, W. Jawaid, F. J. Calero-Nieto, C. Mulas, X. Ibarra-Soria, R. C. V. Tyser, D. L. L. Ho, W. Reik, S. Srinivas, B. D. Simons, J. Nichols, J. C. Marioni, and B. Göttgens, "A single-cell molecular map of mouse gastrulation and early organogenesis," *Nature*, vol. 566, pp. 490–495, Feb 2019.

[12] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[13] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[14] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[15] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[17] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[18] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, p. e00938, 2018.

[19] L. Prechelt, "Early stopping-but when?," *Neural Networks: Tricks of the trade*, pp. 55–69, 1998.

[20] F. Chollet *et al.*, "Keras." `https://keras.io`, 2015.

[21] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

4 Methods

4.1 Dawnn

In contrast to earlier differential abundance detection algorithms, Dawnn has been trained to quantify $\vec{p}$ from simulated datasets. We created simulated training data and learned the relationship between each p and the labels of the 1,000 nearest neighbours of each cell using a deep neural network. This trained model can subsequently be applied to any single-cell RNAseq dataset, for example our test datasets in Table 1.

We tested a number of neural network models in order to identify the best performing architecture. Neural network architectures and the means by which we chose the final one are discussed in the supplementary information (Sections 6.1 and 6.2, respectively).

We assessed six neural network architectures, the effect of dropout layers between each hidden layer and the impact of two learning rates (the amount by which edge weights are updated in each training epoch). The results of the model selection process are shown in Table 2, in which a lower score indicates better performance during training.

Number of nodes per hidden layer	LR = 0.0001 with dropout	LR = 0.0001 no dropout	LR = 0.00001 with dropout	LR = 0.00001 no dropout
[1000]	0.001055	0.001001	0.001855	0.001991
[1000,1000]	0.070437	0.000939	0.071612	0.001094
[250,50,10]	0.008458	0.001107	0.007028	0.001809
[1000,250,50,10]	0.006757	0.000683	0.005723	0.001000
[1000,2000, 1000,250,10]	0.006159	0.00066	0.00787*	0.000654
[1000,2000,4000, 2000,1000,250,10]	0.00676*	0.000708	0.01076*	**0.000617**

Table 2: Results of five-fold cross validation of different neural network architectures, learning rate (LR), and whether dropout. The best-performing algorithm (lowest MSE) is shown in bold. Asterisks indicate incomplete cross validation runs, terminated due to runtime constraints.

We also investigated whether other statistical models – support vector machines [13] and random forests [14] – provide a better basis to estimate $\vec{p}$ (Table 3). The cross-validated mean squared errors for each model exceed those of the better-performing neural networks, indicating worse performance.

Model	Mean squared error
Support vector machine	0.00214
Random forest (100 trees, 50 features)	0.00346
Random forest (500 trees, 50 features)	0.00332
Random forest (100 trees, 250 features)	0.00339
Random forest (500 trees, 250 features)	0.00324

Table 3: Results of five-fold cross validation of different statistical models.

The model selection process identified a neural network with seven hidden layers (containing [1000 2000 4000 2000 1000 250 10] nodes) as having the best performance. This model was trained with a learning rate of 0.00001 and without employing dropout. We then trained this model on the whole training set and incorporated it as the model used by Dawnn.

A more detailed description of neural networks and overview of implementation details of the practical implementations of Dawnn .

4.1.1 Training set generation

Since Dawnn requires training data to learn to estimate $\vec{p}$, there is a risk that good benchmarking performance may only be indicative of similarity between the training and test datasets. If the same assumptions made during training are also satisfied during benchmarking, then good benchmarking performance may not generalise to real data. It is therefore crucial to minimise bias in the training set in order to learn a gener-

alisable model and enable rigorous, objective, and biologically relevant benchmarking. To this end, we ensured that:

1. There is no underlying structure to the cell population of the training set;
2. There is no strong relationship between the labels of neighbouring cells.

If these conditions are satisfied, then we are confident that Dawnn will learn how to estimate $\vec{p}$ in a generalisable manner since the assumptions used when generating the training dataset are so weak.

To satisfy the first condition, we generated each training instance as an independent 1000-cell dataset, rather than by simulating labels on an larger single-cell dataset (which would have introduced assumptions about the dataset's structure).

To satisfy the second condition, we constructed each training instance in a stochastic manner, whereby the p associated with a cell is generated from the p associated with a neighbouring cell by adding a number randomly chosen from a given range, the bounds of which are also randomly chosen for each training instance. The values generated by this process can be thought of as representing the ps of cells along a one-dimensional differentiation trajectory. The pseudocode for the training set generation is given in Algorithm 2. The $\vec{p}$ vectors corresponding to the first ten members of the training set are shown in Figure 6.

Algorithm 2: Training set generation.

1 **for** $j \in \{1, 2, \ldots, 250000\}$ **do**
2 Draw p_1 from Unif$(0, 1)$ `// Starting point`
3 Draw $\hat{s}$ from Unif$(0.001, 0.05)$ `// Maximum step size`
4 **for** $i \in \{2, 3, \ldots, 1001\}$ **do**
5 Draw s from Unif$(-\hat{s}, \hat{s})$
6 $p_i \leftarrow p_{i-1} + s$
7 **if** any p_i not in $[0, 1]$ **then go to** line 2
8 **for** $i \in \{1, 2, \ldots, 1001\}$ **do**
9 **with probability** p_i
10 Assign cell i label S_a
11 **otherwise** Assign cell i label S_b

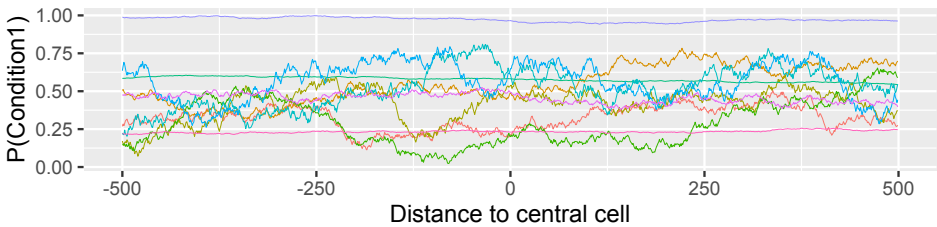


Figure 6: The first ten $\vec{p}$ in the training set. The parameters of the random walk dictate the variation in the distribution across each $\vec{p}$.

Finally, we tested Dawnn in a wide variety of simulated and real-word datasets (Table 1) with different underlying structures to ensure wide applicability to biological problems.

4.2 Benchmarking

4.2.1 DA detection methods

Milo Milo [1] creates overlapping neighbourhoods on a K-nearest neighbour graph and tests each neighbourhood for differential abundance by fitting a negative binomial generalised linear model to its associated counts of different labels. It computes p-values using an F-test. Milo does not natively test for differential abundance at single-cell resolution. We follow Dann *et al.* in converting Milo's neighbourhood level estimates to single-cell resolution by considering "the average outcome in neighborhoods to which a cell belongs" [1].

DA-seq For each cell, DA-seq [2] computes the proportion of cells with a given label within neighbourhoods of different sizes. It fits a logistic regression model to these proportions and uses this model to estimate the probability that the cell belongs to the given sample (i.e. p). It classifies as DA all cells with a more extreme estimated p than the estimates obtained after shuffling labels.

MELD MELD [3] uses a K-nearest neighbour graph to estimate the probability density of each label for each cell (i.e. $\vec{p}$). It does not perform any statistical tests to classify cells as belonging to regions of differential abundance.

Dawnn The algorithm of Dawnn is described in Section 2.1.

4.2.2 Generating Benchmarking Datasets

We employed the benchmarking methodology used by Dann *et al.* when assessing Milo [1], using their code[5] wherever possible. We followed them in using the mouse gastrulation benchmarking dataset and employed their approach to simulate cells in both discrete clusters and in linear and branching trajectories. We used the same approach as for the mouse gastrulation dataset when assigning simulated labels to the keratinocyte, organoid, and heart datasets.

The benchmarking data was simulated under similar statistical assumptions to those made by Dawnn, i.e. that each cell has a probability of being assigned a given label and labels are sampled according to these probabilities. Different benchmarking instances were simulated by first choosing a cluster (corresponding to cell types in the mouse gastrulation dataset) in which the label S_a is maximally up-regulated and then choosing the maximum p (i.e. bias towards S_a) within this cluster. The p each cell is calculated proportionally to its distance from the centre of the chosen up-regulated cluster. Each cell is then assigned the label S_a with probability p and the label S_b otherwise. We made the same choices as Dann *et al.* when selecting both which clusters to exhibit maximal differential abundance and the maximum p in each datasets.

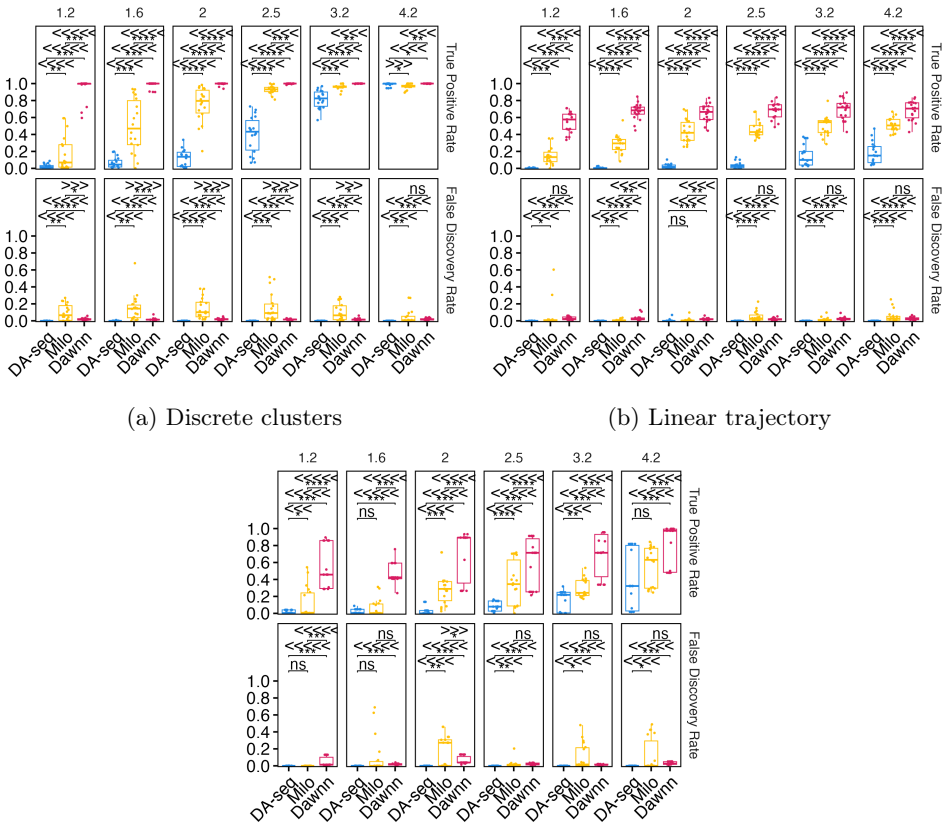
In the ground truth, all cells with p greater than a given threshold are defined as DA and all other cells are not. A threshold of 0.5 was used for the simulated clusters datasets, whilst thresholds of 0.55 where employed everywhere else. Similar thresholds (within 0.005) were used by Dann *et al.* when simulating labels for the mouse gastrulation and simulated linear trajectory datasets.

[5] `https://github.com/MarioniLab/milo_analysis_2020`

Supplementary materials

5 Statistical analyses

Figures 7 and 8 indicate in which benchmarking scenarios there are statistically significant differences between the true positive or false positive rates of Dawnn, Milo, and DA-seq. Results were compared using a paired Wilcoxon signed-rank test with the Bonferroni correction employed to control for multiple comparisons. We measured the effect sizes of the differences as a Z-statistic divided by the square root of the sample size.

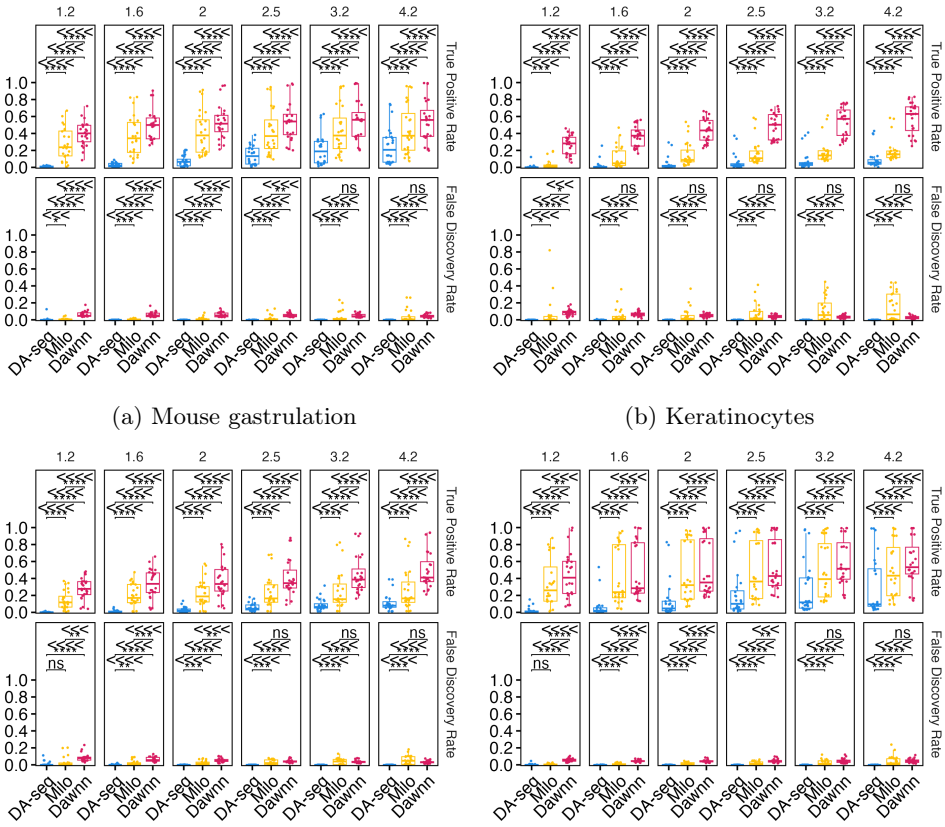


(a) Discrete clusters

(b) Linear trajectory

(c) Branching trajectory

Figure 7: Statistical comparisons of true positive and false discovery rates of DA-seq, Milo, and Dawnn on simulated datasets. The true positive rates of Dawnn are higher than those of Dawnn by a statistically significant extent, which in turn are significantly higher than those of DA-seq. The false discovery rates of Dawnn either do not exhibit a statistically significant difference from those of Milo (b), or are generally significantly smaller (a and c). The FDRs of DA-seq are mostly significantly smaller than those of Dawnn and Milo. For the statistically significant differences, the number of inequality signs show the effect size and the direction of the difference.



(a) Mouse gastrulation

(b) Keratinocytes

(c) Bile duct organoids

(d) Heart

Figure 8: Statistical comparisons of true positive and false discovery rates of DA-seq, Milo, and Dawnn on biological datasets. As with the simulated datasets, the true positive rates of Dawnn are higher than those of Dawnn by a statistically significant extent, and the TPRs of Milo are significantly higher than those of DA-seq. The false discovery rates of Dawnn either do not exhibit a statistically significant difference to or are significantly larger than those of Milo. The FDRs of DA-seq are mostly significantly smaller than those of Dawnn and Milo. For the statistically significant differences, the number of inequality signs show the effect size and the direction of the difference.

6 Machine learning details

6.1 Neural Networks

Neural networks are machine learning models [15, 16] capable of approximating any function [17]. They have been shown to be highly effective in a range of both classification and regression problems [18].

A basic *feed-forward neural network* consists of a sequence of *layers*, each of which contains a collection of *nodes*. Commonly (and as is the case with Dawnn), every node in a layer is connected to every node in the layer following it by an *edge*. Two layers each consisting of 1,000 nodes will therefore have 1,000,000 edges connecting them. Each edge is associated with a numerical *weight*, which approximately signifies its importance within the model. The input (in our case, the labels of the 1,000 nearest neighbours of the cell) is passed layer-by-layer through this network, with each node in a layer taking as input the values at all nodes connected to it in the previous layer. The sum of a node's inputs is then passed through a non-linear function (in our case, the function $\mathrm{ReLU}(x) = \max(0, x)$) and is finally passed along its outgoing edges, with each edge's value being multiplied by its associated weight. A specific neural network architecture is defined by the number of layers, number of nodes in each layer, and configuration of edges connecting the nodes.

6.2 Training a neural network

It is common to train a neural network using the *backpropagation* algorithm [15]. This approach optimises the edge weights of a model to minimise the difference between its current output and the desired values. A neural network has a number of *hyperparameters* that can affect performance, such as its architecture, learning rate (amount by which weights are updated during training), and whether to employ dropout (where edges are deactivated at random during training to prevent overfitting the training data). For five network architectures, we assessed the performance of two learning rates (0.0001 and 0.00001) and the presence of dropout using five-fold cross validation. Cross validation estimates which neural network architectures will best predict $\vec{p}$ in unseen data by training it five times (*folds*), each time on a different 80% of the training data with the remaining 20% used to assess the learned model. Early stopping [19] was employed in each fold. This reduces the risk of overfitting to the training data by halting training once the performance of the model has not improved for some number of training iterations a separate dataset. We terminated the training process after 25 iterations with no improvement.

6.3 Neural network implementation

The neural network at the heart of Dawnn was implemented in the Python libraries Keras [20] and Tensorflow [21]. The architecture selection process resulted in a choice of 1,000 input notes followed by seven dense layers (i.e. where each node in a layer is connected to every node in the preceding one), with these hidden layers containing 2,000, 4,000, 2,000, 1,000, 250, and 10 nodes. We used the ReLU activation function in each layer except the final one (consisting of a single output node), where we use the sigmoid function to constrain the probabilities to $(0, 1)$. Mean squared error was used as the performance metric and Adam was used as the optimiser [22]. To improve generalisation and training efficiency, we used early stopping with a patience of 25 epochs (i.e. training was halted after 25 epochs in which the mean squared error did not decrease). The code used for training Dawnn's neural network is available online[6].

[6] `https://doi.org/10.5522/04/22633606`

7 Label-based representation loses no information

To observe that a label-based representation of a dataset loses no information compared with a proportion-based one, consider the vectors $[S_a \quad S_b \quad S_a \quad S_b]$ and $[S_b \quad S_a \quad S_b \quad S_a]$. Note that both vectors of the proportions of the label S_a ($[1 \quad 0.5 \quad 0.667 \quad 0.5]$ and $[0 \quad 0.5 \quad 0.333 \quad 0.5]$, respectively) are identical at distances 2 and 4. Thus the model inputs at these distances would be identical under a proportion-based representation, whereas they differ under a label-based one. Information can thus be lost when using a proportion-based representation.

8 Parameter values in benchmarking

The tables below contain the values of the parameters used by each algorithm in the benchmarking procedures. The parameter values used by Dann *et al.* for the mouse dataset were used for the other biological datasets since they are of comparable complexity.

Dataset	K	d	prop
Simulated discrete clusters	15	30	0.1
Simulated linear trajectory	20	30	0.1
Simulated branching trajectory	20	30	0.1
Mouse	50	30	0.1
Skin	50	30	0.1
Organoid	50	30	0.1
Heart	50	30	0.1

Table 4: Parameter values of Milo during benchmarking.

Dataset	K vector
Simulated discrete clusters	15, 65, 115, . . . , 465
Simulated linear trajectory	20, 70, 120, . . . , 470
Simulated branching trajectory	50, 100, 150, . . . , 500
Mouse	50, 100, 150, . . . , 500
Skin	50, 100, 150, . . . , 500
Organoid	50, 100, 150, . . . , 500
Heart	50, 100, 150, . . . , 500

Table 5: Parameter values of DA-seq during benchmarking.

Dataset	k	d
Mouse	50	30

Table 6: Parameter values of MELD during benchmarking.

Dataset	alpha	k
All	0.1	1,000

Table 7: Parameter values of Dawnn during benchmarking.

9 Runtime benchmarking results

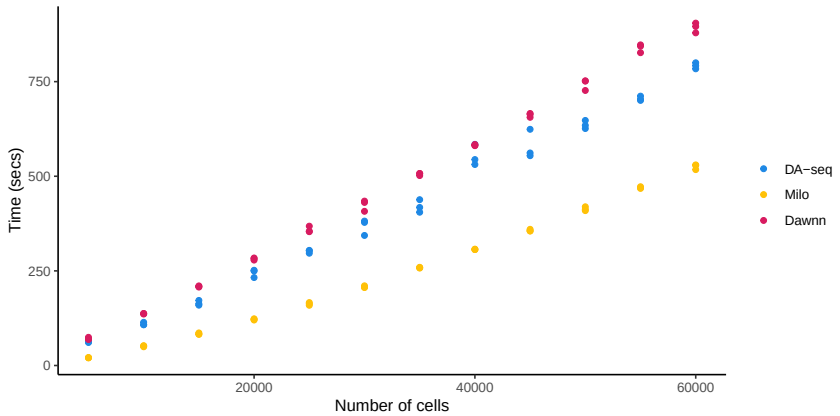


Figure 9: Comparison of runtimes of Dawnn, DA-seq, and Milo on mouse gastrulation dataset. Dataset size (number of cells) is shown on the x axis, runtime (in seconds) is shown on the y axis.