damidBind: an R/Bioconductor package for differential DamID analysis and data exploration

Owen J. Marshall*

January 7, 2026

Abstract

Summary: DamID, and its cell-type specific adaptations, including Targeted DamID (TaDa), Chromatin Accessibility TaDa (CATaDa) and NanoDam, are now widely-adopted as techniques for the genome-wide profiling of DNA binding proteins. Despite this popularity, no software solution exists for identifying differentially bound or accessible loci or transcribed genes between cell types using DamID. The R/Bioconductor package damidBind provides these functions, allowing an end-user to move from raw binding profiles to identifying differentially-bound loci in a reproducible, statistically-robust and straightforward workflow.

Availability and Implementation: damidBind is open-source and freely available from https://bioconductor.org/packages/damidBind/ and https://github.com/marshall-lab/damidBind. The package is implemented in R, and is released under the GPLv3 licence.

Contact: Owen Marshall (owen.marshall@utas.edu.au)

Introduction

DamID is a method for profiling the genome-wide binding of a protein to DNA (van Steensel and Henikoff, 2000; van Steensel *et al.*, 2001). The method fuses a prokaryotic DNA adenine methylase (Dam) to any DNA-binding or -associated protein, resulting in an enrichment of methylated adenines at the sequence GATC at protein binding sites. Methyl-sensitive restriction enzymes combined with ligation-mediated PCR make amplification of methylated regions straightforward, and usable profiles can be obtained from fewer than 10,000 cells (Marshall *et al.*, 2016).

When combined with cell-type specific techniques using bipartite expression systems (as used in Targeted DamID (TaDa) (Southall *et al.*, 2013; Marshall *et al.*, 2016) and Mammalian Targeted DamID (Cheetham *et al.*, 2018; Tosti *et al.*, 2018)), cell-type specific profiles can be obtained from whole organisms or whole tissue, with no cell labelling, cell sorting, or any need for specific antibodies. Now 25 years old, DamID has become widely-adopted, especially since the development of TaDa and its variants.

As well as profiling transcription factors and chromatin proteins, DamID can be used to profile the occupancy of RNA polymerase over gene bodies (Southall *et al.*, 2013). This readout provides a proxy for gene transcription, and has been successfully used for transcriptional profiling of different cell types (e.g. Southall *et al.*, 2013; Marshall and Brand, 2017; Otsuki and Brand, 2018; Doupé *et al.*, 2018; Gervais *et al.*, 2019).

As Dam has a strong affinity for open chromatin, the DamID technique always obtains a Dam-only control profile alongside the Dam-fusion profile, and the final binding profile is a normalised $\log_2$ ratio of binding. The Dam-only binding profile can be used as a direct measure

*Menzies Institute for Medical Research, University of Tasmania, 17 Liverpool St, Hobart 7000, Australia. Correspondence to: owen.marshall@utas.edu.au

of cell-type specifc chromatin accessibility, providing a cell-type specific readout (Chromatin Accessibility TaDa, or CATaDa) similar to ATAC-seq (Aughey *et al.*, 2018).

Multiple software tools and pipelines exist for processing raw DamID-seq NGS reads to $\log_2$ ratio binding profiles and peak calls, including `damidseq_pipeline` (Marshall and Brand, 2015), `Daim` (Tosti *et al.*, 2018) and `damsel` (Page *et al.*, 2024). However, despite the popularity of the TaDa technique, no standardised software package exists to perform downstream differential binding analysis on TaDa, RNA Polymerase TaDa or CATaDa binding profiles generated from different tissues or conditions. Instead, researchers have relied on complex bespoke scripts tailored to individual datasets, and a range of differing analysis solutions.

The R/Bioconductor package `damidBind` formalises the process of performing differential binding analysis on all forms of TaDa datasets. It applies the correct data curation and underlying statistical analysis appropriate for each DamID application, providing a reproducible, statistically-robust and data-appropriate workflow from processed NGS data to biological insights.

Implementation and Methods

`damidBind` is implemented as an R package within the Bioconductor (Gentleman *et al.*, 2004) ecosystem. The primary innovation of `damidBind` lies in providing a unified, reproducible and statistically-appropriate end-to-end analysis pipeline, tailored specifically to DamID data. This includes dedicated functions for parsing, aggregating, and normalising DamID profiles and peaks; consistent and seamless handling of data from the three major DamID-seq experiment types (TaDa, RNA Polymerase TaDa, CATaDa), each with different underlying statistical requirements; and a suite of downstream visualisation and annotation tools that aid understanding of the biological context of results.

A diagrammatic representation of the package workflow is shown in Fig. 1A.

Architecture and input data

`damidBind` is designed for seamless integration within the Bioconductor ecosystem. It uses `GenomicRanges` (Lawrence *et al.*, 2013) objects for genomic data, automatically retrieves up-to-date genome annotations via `AnnotationHub`. Core statistical tests for differential analysis are handled by the robust `limma` (Smyth, 2005) and `NOIseq` (Tarazona *et al.*, 2015) packages, and results are returned in a custom `DamIDResults` S4 object, which provides accessor functions to retrieve the full results analysis table and lists of differentially-bound loci. For downstream functional analysis, the package provides a versatile volcano plot visualisation function, a direct interface to `clusterProfiler` (Yu *et al.*, 2012) for performing Gene Ontology enrichment, and a Shiny web app for interactive visualisation of differentially bound regions in the IGV genome browser (Robinson *et al.*, 2023). `damidBind` makes full use of Bioconductor's `bpparallel` parallelisation methods throughout.

The `damidBind` package assumes that the user has processed NGS files to bedGraph profile tracks and has collected at least two binding replicates per condition. For differential protein binding or chromatin accessibility, the package also requires a predetermined set of peak calls per replicate. `damidBind` accepts two input pathways for these initial processed datasets. A pipeline of the external tools `damidseq_pipeline` (Marshall and Brand, 2015) and `find_peaks` (Marshall *et al.*, 2016) is the current standard within the field and is recommended, with a recent option of using a Snakemake frontend to these software for full data-processing reproducibility and version control (Wit *et al.*, 2025). However, `damidBind` will also accept binding profiles and peak files as a list of `GenomicRanges` R objects. Currently, no solution exists to provide individual normalised binding profiles via an R pipeline, but packages such as `Daim` (Tosti *et al.*, 2018) and `damsel` (Page *et al.*, 2024) provide an alternative means to generate peak calls via R.

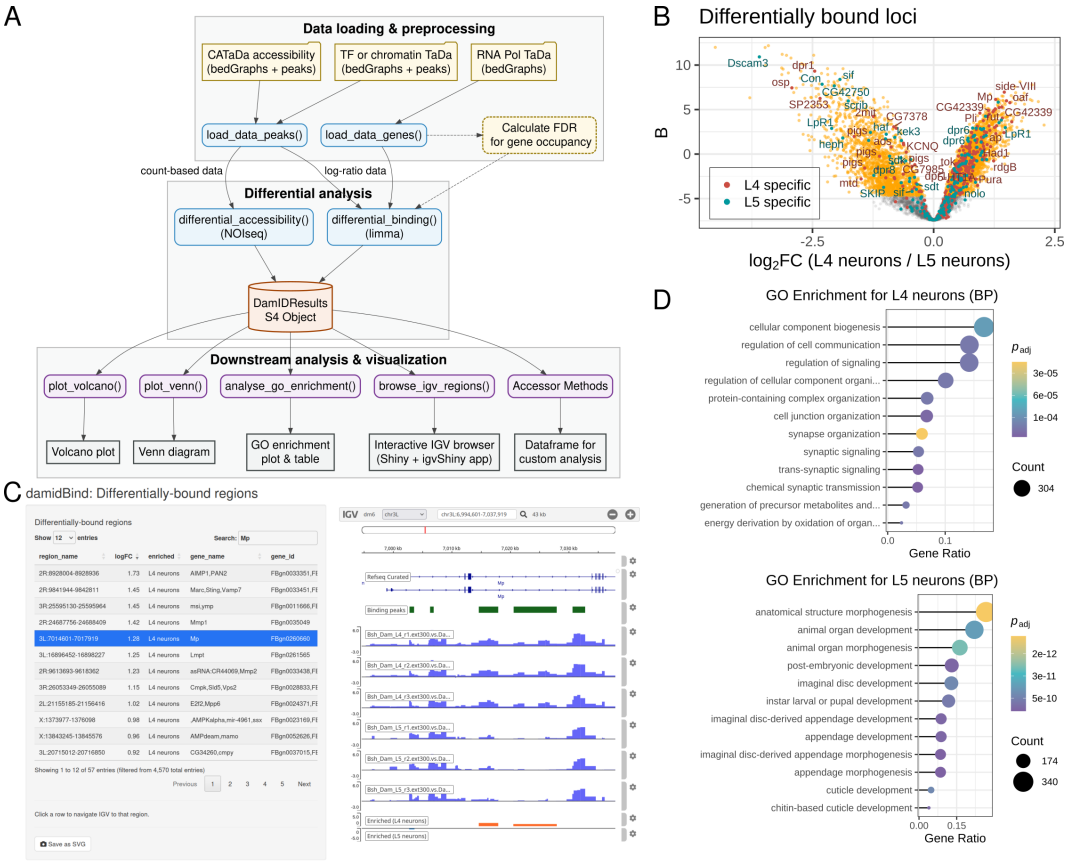

Figure 1: The damidBind package. (A) Schematic overview of the main functions and workflow of damidBind. (B) Example volcano plot of regions differentially bound by the transcription factor Bsh in L4 and L5 lamina neuron subtypes, illustrating gene group highlighting and labelling. (C) A screenshot of the Shiny / IGV based interface for interactive locus browsing. (D) Example GO analysis plot, showing enriched GO terms for genes bound by Bsh in both L4 and L5 neurons.

Core data handling and normalisation

DamID-seq data is a unique form of genomic binding/accesibility data which is unsuitable for differential binding packages designed for ChIP-seq or RNA-seq gene expression analysis. DamID relies on methylated adenines in the sequence GATC, and as a result, the underlying resolution of DamID data is the unevenly-spaced distribution of this sequence in the genome.

At the core of `damidBind`'s analysis is a principle of maximising statistical power by reducing the number of statistical comparisons. For peaks-based analysis pipelines (CATaDa and conventional TaDa), predetermined, per-replicate peak calls are reduced to a set of unified binding regions. For expression analysis, gene body coordinates, as obtained from the most up-to-date genome release (or any historical release) via `AnnotationHub`, are used directly as binding regions.

The variable GATC fragment width within DamID-seq data makes simple count sum and offset analysis inappropriate for differential analysis of these loci. Instead, damidBind calculates the average, fragment-length-weighted signal for each replicate, $\sum(signal_i \times width_i) / \sum(width_i)$ for all fragments i within a locus, where $signal_i$ is either the fragment $\log_2$ enrichment for DamID binding data, or the reads-per-million count for CATaDa. The weighted-mean values per locus are then used in all downstream analysis.

For all workflows, the option of applying quantile normalisation to the input data prior to locus occupancy calculation is provided in order to ensure equivalent data scaling and distribution.

Following data loading, `damidBind` provides diagnostic PCA and correlation heatmap plots of the input data, derived from both the underlying raw data and the mean locus occupancy values, allowing the user to check the data quality, condition separation and batch effects.

Differential analysis workflows

Differential binding of Transcription Factors and other DNA binding proteins

For conventional DamID or cell-type specific Targeted DamID experiments, the `damidBind` workflow is initiated via `load_data_peaks()`. This function imports $\log_2$(Dam-fusion/Dam-alone) binding profiles and associated peak calls, unifies peak regions across all replicates and calculates per-replicate occupancy scores. Peaks are also associated with genes in close proximity (within 2kb of the gene body by default).

The subsequent analysis with `differential_binding()` uses the empirical Bayes framework of `limma` to identify statistically significant changes in protein binding between two conditions. `limma` is designed to handle heteroscedastic $\log_2$-ratio data with a mean-variance trend, as is typical in DamID binding data, and `damidBind` by default uses the `trend` and `robust` options of `limma`'s empirical Bayes moderation to correctly account for this.

Differential gene transcription via RNA Polymerase occupancy

DamID can be used to profile RNA Polymerase occupancy across gene bodies as a proxy for gene expression. `damidBind` directly supports this via the `load_data_genes()` function, which calculates replicate occupancy over annotated genes rather than relying on peak calls. Differential analysis then proceeds using the same `limma`-based `differential_binding()` function as described above.

For RNA Polymerase datasets, `damidBind` provides the ability to determine the FDR of enriched gene body occupancy, as a proxy for gene expression status. The algorithm used here is a substantially revised version of the approach described in Southall *et al.* (2013), and takes advantage of condition replicates to enrich statistical power (see Supplementary Methods for algorithm details). The occupancy FDR values are not used directly in differential expression analysis, but can be used to limit downstream analysis and visualisation plots to expressed genes only.

Differential chromatin accessibility via CATaDa

The Dam-only control sample from a DamID experiment can be used to determine chromatin accessibility, via a method termed CATaDa (Aughey *et al.*, 2018). Unlike $\log_2$ binding ratios, raw CATaDa data is count-based across variable GATC-fragment widths, and `damidBind` provides a dedicated, statistically-appropriate pipeline for handling these data.

Similar to the analysis of protein binding, a set of unified peaks is determined from merging the peak calls of all replicates, and the fragment-length-weighted average signal is calculated per peak, per replicate. The resulting data is a set of continuous, positive numbers that are non-normally distributed, precluding the use of standard tools such as `DESeq2` or `edgeR`, which require integer counts. The `differential_accessibility()` function instead uses `NOIseq` (Tarazona *et al.*, 2015) for differential analysis, a package designed to analyse normalised count-based data non-parametrically, making it robust to this data type. `NOIseq` has an additional advantage in maintaining statistical power over low (two or three) replicate numbers, making it ideal for the typical use cases of CATaDa experiments.

Downstream analysis and visualisation

`damidbind` provides a number of tools for downstream functional interrogation of the results of the `differential_binding()` or `differential_accessibility()` analysis, producing publication-ready plots and figures. These include proportional Venn diagrams of differentially

bound loci using `biovenn` (Hulsen *et al.*, 2008) and Gene Ontology enrichment analysis via `clusterProfiler` (Yu *et al.*, 2012). Because a core feature of `damidBind` is the association of bound loci to neighbouring genes, differential binding analysis is immediately linked to biological function in these analyses.

A highly-customisable volcano plotting function, `plot_volcano()`, allows labelling of loci and the highlighting of any number of gene groups (Fig. 1B). Labelling datapoints in dense plot regions is handled using a custom approach (see Supplementary Methods for algorithm details), enabling clearer identification of significant genes.

With differential binding analysis, it is important to understand the role of bound loci in their surrounding genomic context. For this purpose, `damidBind` also includes an interactive `Shiny` interface around the IGV genome browser (Fig. 1C). All significant differential loci are displayed in an interactive data table, that allows users to quickly sort, search and select results. Clicking on any locus within the datatable displays it in an IGV genome browser frame rendered via `igvShiny`. All replicate tracks, peaks, and differentially bound peaks are shown, along side a display of enrichment magnitude and enrichment direction of differentially bound loci.

Using damidBind to identify differentially-bound and accessible loci within fly lamina neurons

To demonstrate the workflow of `damidBind`, it was applied to a published TaDa dataset profiling the binding of the transcription factor Brain-specific homeobox (Bsh) in two distinct neuronal subtypes of the *Drosophila* optic lobe, L4 and L5 neurons (Xu *et al.*, 2024) (see Supplementary Methods for details).

First, the data were loaded using `load_data_peaks()`, applying quantile normalisation to harmonise the distributions across all samples. The function automatically discovered the relevant bedGraph and peak bed files, merged the binding profiles, reduced the peak sets into a unified list of regions, and annotated these regions with the nearest genes. Next, differential binding analysis was performed with `differential_binding()`, specifying the sample identifiers for each condition (`cond = c("L4 neurons" = "L4", "L5 neurons" = "L5")`). This function returns a `DamIDResults` S4 object, which encapsulates the complete analysis and serves as the input for all downstream functions.

The `plot_volcano()` function provides a visual overview of the differential analysis. Using the `highlight` argument to this function, a list of genes previously identified as specifically expressed in L4 and L5 neurons (Xu *et al.*, 2024) was overlaid, suggesting a strong correlation between neuronal-subtype specific gene expression and Bsh binding solely within the L4 lineage (Fig. 1B). To investigate individual loci, `browse_igv_regions()` provides an interactive Shiny web app containing an IGV browser (Fig. 1C), illustrating enriched Bsh binding at the synaptic signalling gene *Mp (Multiplexin)* in L4 neurons. Finally, to understand the functional implications of these binding dynamics, GO analysis via `analyse_go_enrichment()` revealed enrichment for terms related to synaptic signalling ($p<10^{-4}$) and synapse organisation ($p<10^{-7}$) in L4 neurons, but broad development and morphogenesis GO terms in L5 neurons (Fig. 1D), suggesting different regulatory roles for this transcription factor in different neuronal subtypes.

Conclusions

`damidBind` provides a standardised, comprehensive differential analysis package for DamID-seq data, encompassing all three major uses of the technique, with data diagnostic plots and data visualisation functions that allow researchers to quickly identify key loci for downstream validation and further investigation. The package comes with a comprehensive vignette, and is freely available from Github under the GPLv3 licence. `damidBind` has been accepted into the Bioconductor ecosystem, ensuring continued maintenance and compatibility within the broader

bioinformatics community.

Acknowledgements

I thank Jake Newland and Caroline Delandre for helpful comments on the manuscript, and Andres Wokaty and the Bioconductor community for code review. This work was supported by an National Health and Medical Research Council (NHMRC, Australia) Ideas grant APP2029752.

References

Aughey, G.N, Estacio Gomez, A, Thomson, J, Yin, H and Southall, T.D (2018). CATaDa Reveals Global Remodelling of Chromatin Accessibility during Stem Cell Differentiation in Vivo. *eLife* **7**, 1–22.

Baskerville, G.L (1972). Use of Logarithmic Regression in the Estimation of Plant Biomass. *Can. J. For. Res.* **2**, 49–53.

Benjamini, Y and Hochberg, Y (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Royal Statistical Society. Journal. Series B: Methodological* **57**, 289–300.

Cheetham, S.W, Gruhn, W.H, van den Ameele, J, Krautz, R, Southall, T.D, Kobayashi, T, Surani, M.A and Brand, A.H (2018). Targeted DamID Reveals Differential Binding of Mammalian Pluripotency Factors. *Development* **145**, dev170209.

Doupé, D.P, Marshall, O.J, Dayton, H, Brand, A.H and Perrimon, N (2018). Drosophila Intestinal Stem and Progenitor Cells Are Major Sources and Regulators of Homeostatic Niche Signals. *Proc. Natl. Acad. Sci.* **115**, 12218–12223.

Finney, D.J (1941). On the Distribution of a Variate Whose Logarithm Is Normally Distributed. *Suppl. J. R. Stat. Soc.* **7**, 155–161.

Fisher, R (1925). *Statistical Methods for Research Workers, 11th Ed. Rev* Edinburgh, Oliver and Boyd.

Gentleman, R.C, Carey, V.J, Bates, D.M, Bolstad, B, Dettling, M, Dudoit, S, Ellis, B, Gautier, L, Ge, Y, Gentry, J, Hornik, K, Hothorn, T, Huber, W, Iacus, S, Irizarry, R, Leisch, F, Li, C, Maechler, M, Rossini, A.J, Sawitzki, G, Smith, C, Smyth, G, Tierney, L, Yang, J.Y and Zhang, J (2004). Bioconductor: Open Software Development for Computational Biology and Bioinformatics. *Genome Biol* **5**, R80.

Gervais, L, van den Beek, M, Josserand, M, Sallé, J, Stefanutti, M, Perdigoto, C.N, Skorski, P, Mazouni, K, Marshall, O.J, Brand, A.H, Schweisguth, F and Bardin, A.J (2019). Stem Cell Proliferation Is Kept in Check by the Chromatin Regulators Kismet/CHD7/CHD8 and Trr/MLL3/4. *Dev. Cell* **49**, 556–573.

Hulsen, T, de Vlieg, J and Alkema, W (2008). BioVenn - a Web Application for the Comparison and Visualization of Biological Lists Using Area-Proportional Venn Diagrams. *BMC genomics* **9**, 488.

Lawrence, M, Huber, W, Pagès, H, Aboyoun, P, Carlson, M, Gentleman, R, Morgan, M.T and Carey, V.J (2013). Software for Computing and Annotating Genomic Ranges. *PLoS Comput Biol* **9**, e1003118.

Marshall, O.J and Brand, A.H (2015). Damidseq_pipeline: An Automated Pipeline for Processing DamID Sequencing Datasets. *Bioinformatics* **31**, 3371–3.

Marshall, O.J and Brand, A.H (2017). Chromatin State Changes during Neural Development Revealed by in Vivo Cell-Type Specific Profiling. *Nat. Commun.* **8**, 2271.

Marshall, O.J, Southall, T.D, Cheetham, S.W and Brand, A.H (2016). Cell-Type-Specific Profiling of Protein–DNA Interactions without Cell Isolation Using Targeted DamID with next-Generation Sequencing. *Nat. Protoc.* **11**, 1586–1598.

Otsuki, L and Brand, A.H (2018). Cell Cycle Heterogeneity Directs the Timing of Neural Stem Cell Activation from Quiescence. *Science* **360**, 99–102.

Page, C.G, Lonsdale, A, Mitchell, K.A, Schröder, J, Harvey, K.F and Oshlack, A (2024). Damsel: Analysis and Visualisation of DamID Sequencing in R. *Bioinformatics* **40**, btae695.

Robinson, J.T, Thorvaldsdottir, H, Turner, D and Mesirov, J.P (2023). Igv.Js: An Embeddable JavaScript Implementation of the Integrative Genomics Viewer (IGV). *Bioinformatics* **39**, btac830.

Smyth, G.K (2005). Limma: Linear Models for Microarray Data. In Gentleman, R, Carey,

V.J, Huber, W, Irizarry, R.A and Dudoit, S (eds), *Bioinformatics and Computational Biology Solutions Using R and Bioconductor* Springer, New York, NY , pp. 397–420.

Southall, T.D, Gold, K.S, Egger, B, Davidson, C.M, Caygill, E.E, Marshall, O.J and Brand, A.H (2013). Cell-Type-Specific Profiling of Gene Expression and Chromatin Binding without Cell Isolation: Assaying RNA Pol II Occupancy in Neural Stem Cells. *Dev. Cell* **26**, 101–12.

Stouffer, S.A, Suchman, E.A, Devinney, L.C, Star, S.A and Williams Jr., R.M (1949). *The American Soldier: Adjustment during Army Life. (Studies in Social Psychology in World War II), Vol. 1* Princeton Univ. Press, Oxford, England.

Tarazona, S, Furió-Tarí, P, Turrà, D, Di Pietro, A, Nueda, M.J, Ferrer, A and Conesa, A (2015). Data Quality Aware Analysis of Differential Expression in RNA-seq with NOISeq R/Bioc Package. *Nucleic Acids Res.* **43**.

Tosti, L, Ashmore, J, Tan, B.S.N, Carbone, B, Mistri, T.K, Wilson, V, Tomlinson, S.R and Kaji, K (2018). Mapping Transcription Factor Occupancy Using Minimal Numbers of Cells in Vitro and in Vivo. *Genome Res.* **28**, 592–605.

van Steensel, B and Henikoff, S (2000). Identification of in Vivo DNA Targets of Chromatin Proteins Using Tethered Dam Methyltransferase. *Nat. Biotechnol.* **18**, 424–8.

van Steensel, B, Delrow, J and Henikoff, S (2001). Chromatin Profiling Using Targeted DNA Adenine Methyltransferase. *Nat. Genet.* **27**, 304–8.

Wit, N, Bertlin, J, Hynes-Allen, A, van den Ameele, J and Nathan, J (2025). Mapping SET1B Chromatin Interactions with DamID Using DamMapper, a Comprehensive Snakemake Workflow. *BMC Genomics* **26**, 914.

Xu, C, Ramos, T.B, Marshall, O.J and Doe, C.Q (2024). Notch Signaling and Bsh Homeodomain Activity Are Integrated to Diversify Drosophila Lamina Neuron Types. *eLife* **12**, RP90136.

Yu, G, Wang, L.G, Han, Y and He, Q.Y (2012). ClusterProfiler: An R Package for Comparing Biological Themes among Gene Clusters. *OMICS J. Integr. Biol.* **16**, 284–287.

Supplementary material

DamID-seq data sources

Binding data for Bsh (Xu *et al.*, 2024) was obtained from GEO accession number GSE247239, processed with `damidseq_pipeline` v1.6.3 and `find_peaks` v1.2 with default settings. RNA Polymerase occupancy data (Marshall and Brand, 2017) was obtained from GEO accession number GSE77860, processed with `damidseq_pipeline` v1.6.3 with default settings.

The processed data and peak calls are provided in a Zenodo dataset at doi.org/10.5281/zenodo.16649477. This processed dataset consists of $\log_2$-ratio binding profiles, alongside associated peak calls for the Bsh data,

Deeper labelling of dense point clouds on volcano plots

Differential analysis typically involves the display of hundreds to thousands of significant loci in volcano plots. Current graphical label packages (e.g. `ggrepel` on thousands of datapoints) within the R ecosystem are limited by total non-overlapping label plot space constraints, meaning that significant point clusters with high density remain unlabelled.

To solve this, `damidBind` uses Algorithm 1 to thin the list of labelled points in dense regions, displaying the thinned list with `ggrepel`. The result is deeper labelling within the significant loci cloud, increasing the number of loci with displayed labels within dense regions of volcano plots, while still preserving labels on isolated points.

Algorithm 1: Sampling points by local isolation

Input : D: A dataframe containing coordinates
k: An integer for both search depth and priority calculation
r: A fixed numeric exclusion radius
Output: V_{kept}: A logical vector indicating points selected for labelling

```
# Pre-process, calculate isolation priority and sort
```

$X \leftarrow$ Extract (x, y) coordinates from D
$X_{scaled} \leftarrow$ Standardise X (centre and scale to unit variance)
$N_{dist} \leftarrow$ For each point in X_{scaled}, find distances to its k nearest neighbours
$P \leftarrow$ Calculate the median of N_{dist} for each point $S \leftarrow$ Indices of points sorted by P in descending order

```
# Iterative thinning via radius-based exclusion
```

Initialise $V_{processed}$ as a boolean vector of size $|D|$ set to *False*
Initialise V_{kept} as a boolean vector of size $|D|$ set to *False*
foreach *index* $i \in S$ **do**
  if $V_{processed}[i]$ *is False* **then**
    $V_{kept}[i] \leftarrow$ *True*
    $V_{processed}[i] \leftarrow$ *True*
    $R_{idx} \leftarrow$ Indices of all points in X_{scaled} within distance r of $X_{scaled}[i]$
    foreach *index* $j \in R_{idx}$ **do**
      $V_{processed}[j] \leftarrow$ *True*
    end
  end
end
return V_{kept}

Determining gene occupancy FDR

While `damidBind` is primarily concerned with differential analysis, RNA Polymerase DamID, as a proxy for gene expression, allows an additional option of restricting any downstream visualisation to expressed genes only. The gene body occupancy FDR algorithm used for this purpose by `damidBind` is substantially modified from the original described in Southall *et al.* (2013), using a two-tiered regression modelling approach built upon a permutation-based null distribution. The objective is to create a predictive model (Algorithm 2) that can estimate an empirical p-value for any given gene's observed occupancy score and fragment count (Algorithm 3). The per-gene p-values from each replicate are grouped by test condition during the `differential_binding()` call and aggregated using either Fisher's (Fisher, 1925) or Stouffer's (Stouffer *et al.*, 1949) method, to give a per-condition p-value for each gene. Finally, these are adjusted for multiple hypothesis testing using the Benjamini-Hochberg method (Benjamini and Hochberg, 1995) to yield a final per-gene occupancy FDR.

When applied to correctly-normalised RNA Polymerase DamID data, the occupancy FDR serves as a proxy for gene expression (Southall *et al.*, 2013), although genes with significant paused RNA Polymerase may also be detected using this approach. The gene occupancy FDR values are not directly used to determine differential gene expression changes, but are used to filter the downstream plots and analysis for expressed genes only.

Major changes from Southall *et al.* (2013) include:

1. The use of fragment-length-weighted mean values for building the null distribution.
2. Generating the underlying null model distributions by sampling with replacement.
3. The use of log-linear models for the Tier 1 regressions.
4. Using natural spline fits for the Tier 2 regressions (slope, intercept, mean squared error (MSE) of residuals) to capture non-linear relationships across the range of fragment counts.
5. The use of weighted least squares (WLS) for Tier 2 slope and intercept models, using the inverse of the Tier 1 standard errors as weights, to account for model heteroscedasticity (Baskerville, 1972).
6. Applying a bias correction factor derived from the predicted MSE when back-transforming from the log-scale, to correct for Jensen's inequality (Finney, 1941; Baskerville, 1972).
7. Determining a condition-level FDR by combining unadjusted replicate p-values using either Stouffer's Z-score method or Fisher's method, followed by Benjamini-Hochberg (BH) adjustment at the condition level.

In order to capture the non-linear relationship between fragment counts and the regression parameters for the Tier 2 regressions, the slope and intercept are modelled as a function of the log-transformed fragment count (Fig. S1A, B). The pseudo-linear residual variance (MSE) is modelled on the linear fragment count (Fig. S1C) to accurately represent the variance profile in low fragment count genes.

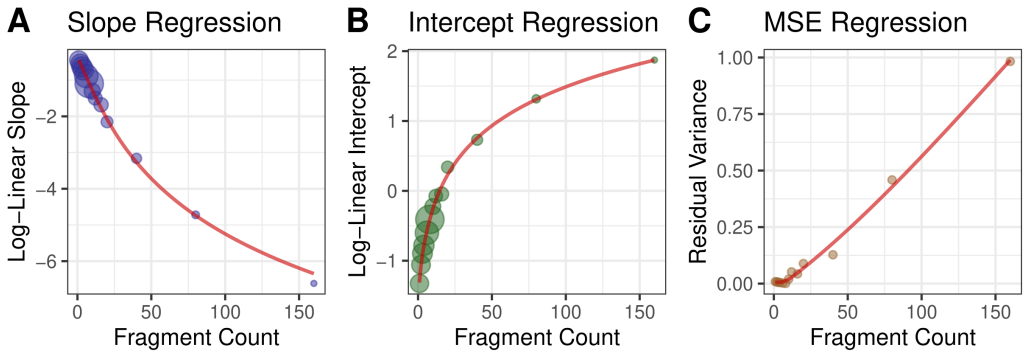

Figure S1: FDR modelling for RNA Polymerase occupancy data. The plots show the Tier 2 regression fits and Tier 1 model heteroscedasticity, for a typical DamID RNA Polymerase dataset. Both the Tier 2 slope (A) and intercept (B) regressions use weighted natural spline fits to fit the underlying data; circle size is proportional to weight. The model mean squared error (MSE) increases with fragment count (C) and is also fitted with an unweighted natural spline to apply the appropriate model bias correction upon back-transformation. Data illustrated is derived from RNA Polymerase (RpII215) occupancy data from GSE77860.

Algorithm 2: Occupancy FDR: Model training

Input : S_{genome}: Binding profile scores
W_{genome}: Fragment widths
N_{iter}: Simulation iterations
C_{frag}: Set of fragment counts
T_{occ}: Set of occupancy thresholds
Output: Models M_{slope}, M_{int}, and M_{MSE}

```
# Simulate null distribution by sampling fragments
```
Initialise matrix H of size $|T_{\text{occ}}| \times |C_{\text{frag}}|$ with zeros
for $n \leftarrow 1$ **to** N_{iter} **do**
  foreach $f \in C_{\text{frag}}$ **do**
    Idx $\leftarrow$ Sample f indices from profile with replacement
    $\text{occ}_{\text{rand}} \leftarrow \frac{\sum_{i \in \text{Idx}} S_{\text{genome}}[i] \cdot W_{\text{genome}}[i]}{\sum_{i \in \text{Idx}} W_{\text{genome}}[i]}$ `# Weighted mean of GATC fragments`
    foreach $t \in T_{\text{occ}}$ **do**
      if $\text{occ}_{\text{rand}} > t$ **then**
        $H[t, f] \leftarrow H[t, f] + 1$
      end
    end
  end
end
$P \leftarrow H / N_{\text{iter}}$ `# Empirical probabilities`

```
# Tier 1: Log-linear regression for each fragment count
```
foreach $f \in C_{\text{frag}}$ **do**
  $P_f \leftarrow$ Probability column for f; $T_f \leftarrow T_{\text{occ}}$
  Remove indices where $P_f = 0$
  if remaining indices ≥ 3 **then**
    Fit M_f: $\log(P_f) \sim T_f$
    Extract coefficients ($\beta_{\text{slope},f}, \beta_{\text{int},f}$), standard errors ($\text{SE}_{\text{slope},f}, \text{SE}_{\text{int},f}$), and residual variance (MSE_f)
  end
end

```
# Tier 2: Weighted natural spline (NS) regressions
```
$f_{\log} \leftarrow \log(f)$
Fit M_{slope}: $\beta_{\text{slope}} \sim \text{NS}(f_{\log}, \text{df} = 3)$ with weights $1/(\text{SE}_{\text{slope}} + \epsilon)$
Fit M_{int}: $\beta_{\text{int}} \sim \text{NS}(f_{\log}, \text{df} = 3)$ with weights $1/(\text{SE}_{\text{int}} + \epsilon)$
Fit M_{MSE}: $\text{MSE} \sim \text{NS}(f, \text{df} = 3)$ `# Unweighted`
return $M_{\text{slope}}, M_{\text{int}}, M_{\text{MSE}}$

Algorithm 3: Occupancy FDR: p-value estimates and replicate integration

Input : G: Set of genes with observed occupancy and fragment counts per replicate
C: Set of test conditions linked to replicates r
$M_{\text{slope},r}, M_{\text{int},r}, M_{\text{MSE},r}$: Pre-trained Tier 2 models for each replicate r
Method: Statistical approach (Stouffer or Fisher)
Output: G with condition-level adjusted $g.\text{FDR}_c$ values

```
# Estimate individual replicate p-values
foreach condition c ∈ C do
    foreach replicate r ∈ c do
        foreach gene g ∈ G do
            if g.nfrag > 0 then
                pred_slope ← Predict from M_slope,r using log(g.nfrag)
                pred_int ← Predict from M_int,r using log(g.nfrag)
                pred_MSE ← max(0, Predict from M_MSE,r using g.nfrag)
                log(p) ← (pred_slope × g.occ) + pred_int
                p_g,r ← exp(log(p) + pred_MSE/2)          # Jensen's correction
                p_g,r ← clamp(p_g,r, min = 10^-16, max = 1.0)
            end
            else
                p_g,r ← 1.0
            end
        end
    end

    # Aggregate p-values across replicates per condition
    foreach gene g ∈ G do
        k ← Count of replicates for g where p_g,r < 1.0
        if k = 1 then
            P_combined,g ← the single non-1.0 p-value
        end
        else if Method is Stouffer then
            foreach r ∈ c do
                z_r ← Φ^-1(1 − p_g,r)                     # Inverse Normal CDF
            end
            Z_meta ← (Σ z_r) / √k                         # Unweighted combination
            P_combined,g ← 1 − Φ(Z_meta)                  # Back-transform
        end
        else if Method is Fisher then
            X^2 ← −2 Σ_{r=1}^{k} log(p_g,r)               # Fisher statistic
            P_combined,g ← P(χ^2_{2k} > X^2)              # Chi-squared lookup
        end
    end

    # FDR correction at condition level
    G.FDR_c ← Benjamini-Hochberg(P_combined)
    return G with assigned G.FDR_c
end
```