nature communications

Article

https://doi.org/10.1038/s41467-025-64383-1

Computational design and evaluation of optimal bait sets for scalable proximity proteomics

Received: 10 October 2024

Accepted: 15 September 2025

Published online: 22 October 2025

Check for updates

Vesal Kasmaeifar[1,2], Saya Sedighi[1,2], Anne-Claude Gingras[1,2] & Kieran R. Campbell[1,2,3,4,5,6]

The spatial organization of proteins within eukaryotic cells underlies essential biological processes and can be mapped by identifying nearby proteins using proximity-dependent biotinylation approaches such as BioID. When applied systematically to hundreds of bait proteins, BioID has localized thousands of endogenous proteins in human cells, generating a comprehensive view of subcellular organization. However, the need for large bait sets limits the scalability of BioID for context-dependent spatial profiling across different cell types, states, or perturbations. To address this, we develop a benchmarking framework with multiple complementary metrics to assess how well a given bait subset recapitulates the structure and coverage of a reference BioID dataset. We also introduce GENBAIT, a genetic algorithm-based method that identifies optimized bait subsets predicted to retain maximal spatial information while reducing the total number of baits. Applied to three large BioID datasets, GENBAIT consistently selected subsets representing less than one-third of the original baits while preserving high coverage and network integrity. This flexible, data-driven approach enables intelligent bait selection for targeted, context-specific studies, thereby expanding the accessibility of large-scale subcellular proteome mapping.

Spatial partitioning is fundamental to biological systems, from organs to cells and their subcellular compartments[1,2]. Eukaryotic cells organize biochemical activities into distinct structures, enabling proteins to function in specific environments. Disruptions in localization can lead to disease, underscoring the importance of mapping protein spatial organization across biological contexts[3,4].

Several strategies have been developed to chart protein localization, including fluorescence microscopy[3], biochemical fractionation coupled with mass spectrometry (MS)[5,6], cross-linking MS[7], and computational modeling[8]. While useful for studying most cellular components, fluorescence-based imaging requires assessing the localization of each protein individually, making proteome-wide, multi-condition studies labor-intensive[3]. Biochemical fractionation coupled with MS is more scalable and has delineated numerous cellular proteomes and sub-proteomes[4,6,9,10], but is less effective for structures that are difficult to isolate, such as most membraneless organelles[11].

Proximity-dependent biotinylation (or proximity labeling, PL) approaches, such as BioID, APEX, and their derivatives, overcome some of these limitations by capturing proteins in the immediate molecular neighborhood of a protein of interest in living cells[12]. In these methods, a labeling enzyme is genetically fused to a protein of interest (known as the bait) and expressed in a relevant cellular context. BioID experiments typically involve exogenous expression of the bait, which ensures sufficient expression even for proteins with low

[1]Lunenfeld-Tanenbaum Research Institute, Mount Sinai Hospital, Sinai Health, Toronto, ON, Canada. [2]Department of Molecular Genetics, University of Toronto, Toronto, ON, Canada. [3]Department of Statistical Sciences, University of Toronto, Toronto, ON, Canada. [4]Department of Computer Science, University of Toronto, Toronto, ON, Canada. [5]Ontario Institute of Cancer Research, Toronto, ON, Canada. [6]Vector Institute, Toronto, ON, Canada. e-mail: gingras@lunenfeld.ca; kierancampbell@lunenfeld.ca

endogenous levels and allows targeted labeling using protein fragments or subcellular localization signals[13]. Subsequent addition of the substrate of the enzyme results in the covalent labeling of adjacent proteins (known as preys) in living cells (Fig. 1a). In BioID[14], an abortive biotin ligase produces reactive biotinoyl-AMP that covalently modifies the lysine ε-amines of proteins within ~10 nm[15]. The biotinylated proteins are then purified using streptavidin and identified by MS. Protein identifications are scored against negative controls (e.g., using significance analysis of interactome (SAINT)[16–18]) to identify high-confidence proximal interactions and generate spatially resolved proteomic data. BioID has been widely used to characterize proximal interactions for multiple individual baits, across cellular compartments, cell types, and even organisms[12,19]. Note, however, that labeling efficiency can vary across cellular compartments[20,21].

Beyond mapping individual bait–prey interactions, the richness of large-scale BioID datasets also enables prey-centric analyses. By analyzing prey co-occurrence patterns using methods such as correlation analysis or non-negative matrix factorization (NMF)[22], it is possible to group proteins with similar proximity profiles, revealing likely co-localization. These data can then be used to reconstruct cellular (or subcellular) organization[4,23]. NMF is a linear dimensionality reduction technique that decomposes a bait–prey matrix into a basis matrix and a score matrix. It is well-suited for defining organellar composition because it soft-assigns prey proteins into a pre-specified number of components, simultaneously reflecting the cellular reality, in which many proteins have no singly defined subcellular localization. While the user is free to set the number of components, a heuristic in g:Profiler[24] based on Gene Ontology Cellular Component (GO:CC) analysis of the preys assigned to each component is commonly used[4,23]. Components identities are annotated following a guilt-by-association principle, using GO terms or similar reference sets.

This workflow has produced BioID-based maps of the human cell[4], cytosolic mRNA-associated granules and bodies[23], nuclear bodies[25], mitochondria[26], and centrosome-cilium interface[27]. However, these studies were generally performed in a single cell line under constant growth conditions, primarily due to the labor and cost of profiling large bait sets. Each large-scale BioID map can require months to years of work, substantial reagent costs, and extensive MS time, making similar mapping across multiple conditions impractical.

A means to select bait subsets that can recapitulate the structures of these proximal interactomes with similar prey coverage would enable subcellular protein localization mapping in different contexts while reducing costs and experimental burden in direct proportion to the reduction in bait numbers. From a machine learning perspective, this can be framed as a feature selection problem[28], in which a subset of features (i.e., baits) is selected that maintains the strong predictive capacity of the full set. While there are a multitude of algorithms for feature selection that have been extensively applied to problems in biological data analysis[29,30], none have been evaluated for their ability to generate bait subsets for scalable BioID profiling studies using formalized metrics, and no bespoke approaches have been proposed.

In this work, we develop GENBAIT, a genetic algorithm-based strategy for BioID bait subset selection and introduce a benchmarking platform to quantify the quality of the BioID bait subsets, which we use to compare GENBAIT's performance with that of 10 existing statistical and machine learning-based feature selection algorithms. While all selection methods performed markedly better than random subset selection, GENBAIT outperformed the other methods across several metrics. Additionally, we formalize a set of recommendations to help researchers choose the optimal method to derive bait subsets from existing BioID datasets for scalable subcellular profiling. By reducing the number of baits required for large-scale BioID experiments, GENBAIT will enable studies that would otherwise be infeasible due to cost and time constraints. GENBAIT is not intended to replace highly targeted approaches but rather to provide a systematic and data-driven strategy for bait selection when a broad spatial proteomics map is needed. A Python package implementing both GENBAIT and the 15 metrics for assessing bait subset quality is available at https://github.com/camlab-bioml/genbait.

Results

A computational platform to design and evaluate BioID bait subsets

To evaluate the quality of bait subsets selected by feature selection methods, we used three available BioID datasets, all acquired in HEK-293 Flp-In T-REx 293 cells but representing distinct biological contexts. Dataset 1, the Human Cell Map V.1 (humancellmap.org), provides a global reference for intracellular organization, featuring 192 baits and 4145 high-confidence preys, mapping major organelles, cytoskeletal structures and some membraneless organelles. Dataset 2 focuses on stress granules and P-bodies, along with other components of the RNA synthesis, transport and degradation machineries, using 119 baits and 1792 preys. Dataset 3 maps nuclear bodies (including nucleolus, nuclear speckles, paraspeckles), leveraging 140 baits and 1816 preys. These datasets differ in terms of their coverage and resolution, with varying numbers of baits and preys across cellular structures (Fig. 1a). NMF optimization in the original publications defined 20, 14 and 19 components, respectively, for these datasets. We further developed a benchmarking resource comprising 15 different evaluation metrics that quantified how well a given subset of baits captures the localizations determined in the original dataset and applied this to all BioID datasets individually (Fig. 1b).

Next, we developed a computational method to select bait subsets across multiple BioID experiments that can reproduce the interaction network of the full dataset, enabling scalable profiling. Given their success at solving similar discrete optimization problems[31–35], we adopted a genetic algorithm-based search strategy informed by the typical BioID data analysis workflow. Implementing this strategy requires specifying a fitness function that quantifies how well a subset conserved the subcellular localizations observed with the full bait set. We developed a fitness function that mimics the iterative data analysis process[4,23] to interpret subcellular colocalizations from BioID data (Fig. 1c). Specifically, for a proposed bait set, we applied NMF to the bait–prey matrix, retaining only the proposed bait subsets with the same number of components as the original NMF fit. We then used the Hungarian algorithm[36] to re-order the components to best match the original and computed each component's Pearson correlation between the original and subset NMF components. Finally, a single fitness score was computed by averaging the correlations of the best-matched components between the original and subset, with a penalty on low-value components to ensure comprehensive component capture ("Methods").

Starting with an initial set of randomly generated bait subsets, the algorithm iteratively applied crossover (swapping parts of two or more bait subsets to create new ones), and mutation (randomly changing baits in the bait subsets to introduce variability) operations[37] to generate new subsets, then evaluated each with a fitness function. The subsets with the highest fitness scores were selected to generate subsequent subsets, and the operations were repeated. Over successive generations, this process generated optimized bait subsets for scalable BioID profiling (Fig. 1d).

Contrasting bait subset selection by GENBAIT to random selection

To compare the efficacy of a GENBAIT-determined bait subset to a randomly selected subset of the same size, we selected ~1/3 of the baits in each dataset (60, 40, and 45 for datasets 1, 2, and 3, respectively) as a reasonable subset size, intended to approximate a moderate fraction of the total bait set. We then recomputed their NMF representations

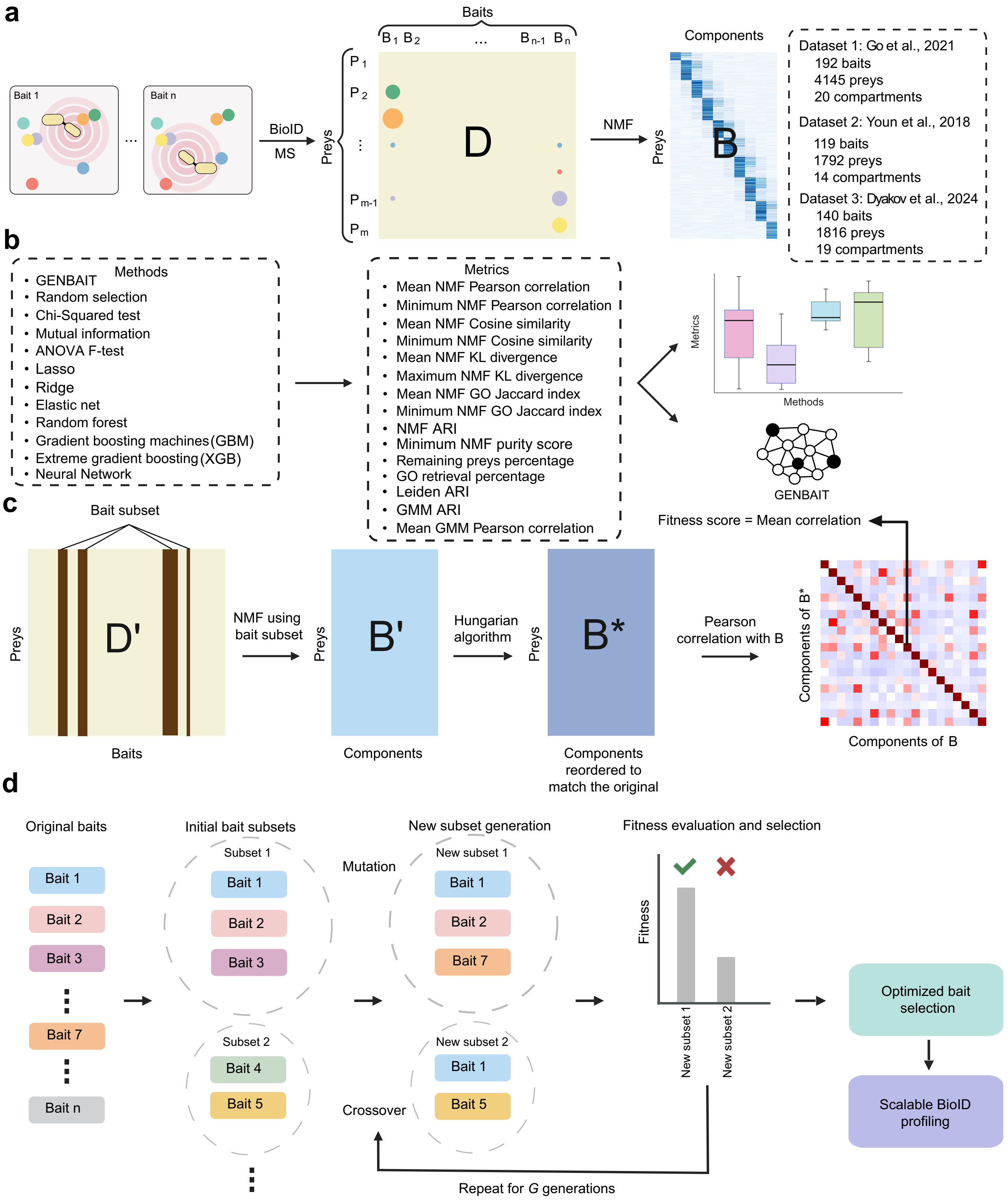

Fig. 1 | GENBAIT workflow and evaluation. a Proximity labeling data are acquired by MS. Interaction scoring is performed to generate a matrix of baits and high-confidence preys D. NMF is used to soft-assign preys into a predefined number of components B based on GO:CC terms. **b** We compared GENBAIT's performance to 10 feature selection methods and random selection (as a baseline) using 15 different metrics. **c** Schematic of the evaluation procedure (fitness function). Bait selection generates a subset D′ of the original dataset D. NMF is used to soft assign the subset's preys into components B′, and the Hungarian algorithm is used to create a matrix B* in which the components of B′ are aligned with those in the full dataset B. Then, Pearson correlations between corresponding components are calculated. The mean of the diagonal values is used as the fitness score in the genetic algorithm. **d** Workflow of the genetic algorithm for optimizing BioID bait subsets. Randomly selected initial bait subsets undergo mutation and crossover operations, followed by fitness evaluation. High-scoring subsets are used in the next generation to iteratively define the optimal subset for scalable BioID profiling.

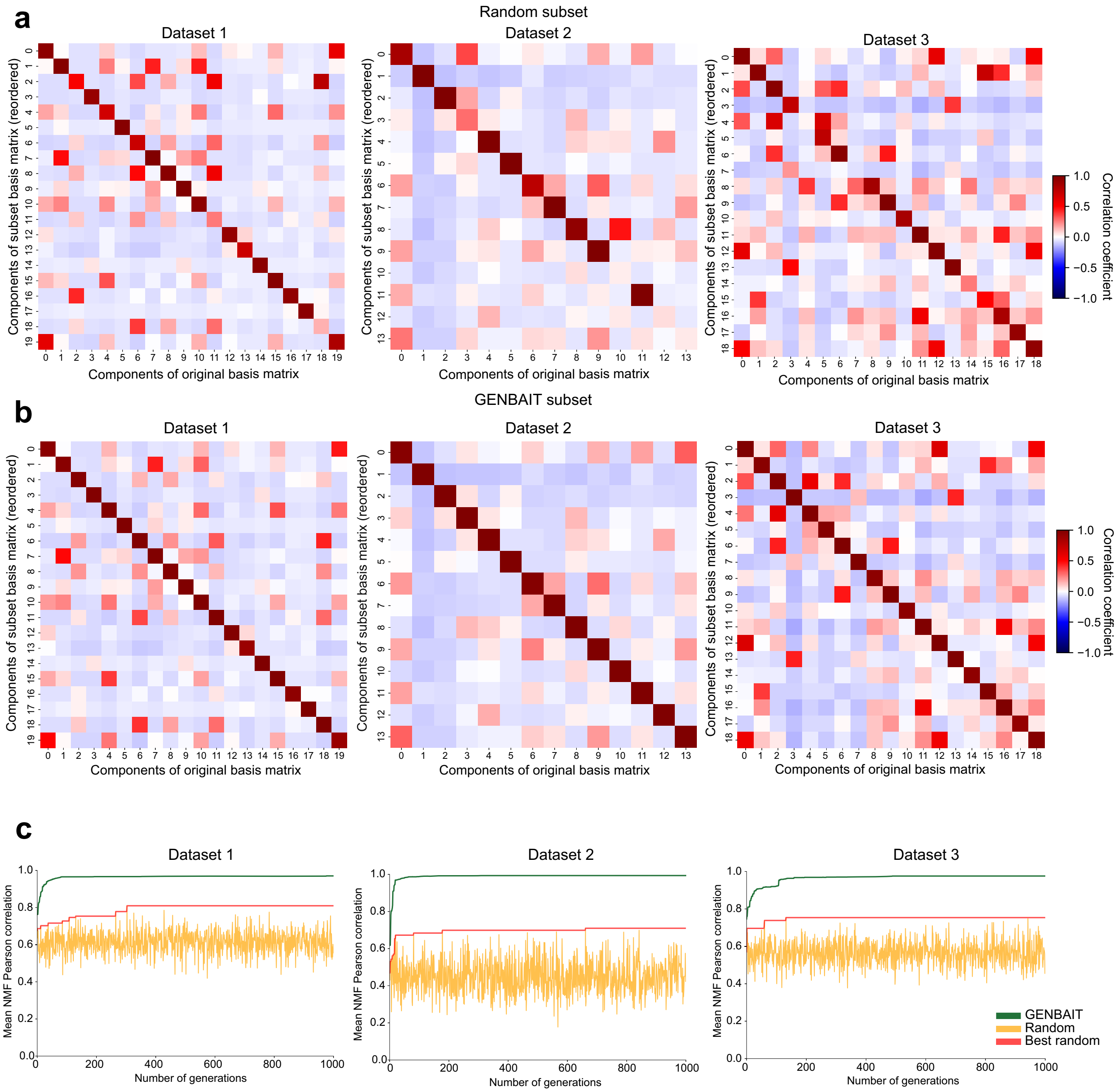

Fig. 2 | Comparing bait selection with GENBAIT against random selection. **a**, **b** Correlation heatmaps between the original NMF results across all datasets and randomly selected subsets of each. Diagonal values indicate correlations between corresponding components of the original and subset basis matrices. **c** Comparison of mean NMF Pearson correlation scores over 1000 generations of bait subsets using either GENBAIT or random selection or best random subset.

and measured their correlations with the NMF results of the full datasets (Fig. 2a).

While these random subsets captured most components (as shown by the high correlation coefficients on the heatmap diagonals), they missed important components in all datasets. For dataset 1, components related to the cytoskeleton, endoplasmic reticulum membrane, and nucleus showed low correlation values. Similarly, for dataset 2, components related to the endoplasmic reticulum membrane, cytoskeleton, spliceosomal complex, and nucleoplasm had lower correlations with the original dataset. In dataset 3, components corresponding to the endoplasmic reticulum, cytoskeleton, microtubule, and spliceosomal complex showed low correlation values, indicating that these regions were not well preserved in random subsets. Notably, when GENBAIT was used to generate optimized bait subsets, all the components showed remarkably higher correlation values (Fig. 2b). We also compared GENBAIT to a heuristic approach that selects baits based on the number of preys they capture (Supplementary Fig. 1). This method resulted in lower correlation values for several components. GENBAIT, however, maintained higher correlations across all datasets, further demonstrating its effectiveness in optimizing bait selection for proximity proteomics.

Since GENBAIT iteratively generated 1000 bait subsets, we investigated whether random selection could achieve a similarly high-scoring subset given the same computational resources. We compared mean NMF Pearson correlation scores of GENBAIT-selected subsets to those of random subsets generated at each iteration, also tracking the highest-scoring random subset across all iterations. GENBAIT quickly reached a peak mean NMF Pearson correlation score (<100 generations), whereas random subsets, including the best-performing one, failed to do so even after 1000 generations (Fig. 2c).

Comprehensive benchmarking across multiple feature selection methods and metrics

We next used our benchmarking pipeline to quantify the performances of GENBAIT and 10 other feature selection methods (Fig. 1b and Supplementary Table 1), including statistical tests of the associations between baits and components (e.g., analysis of variance (ANOVA F), mutual information and Chi-squared)[38], variants of sparse regression that select baits with non-zero coefficients for the subset (lasso, ridge, and elasticnet)[39,40], and tests that measure the feature importance per bait (random forest, gradient boosting machines (GBM)[41,42] and extreme gradient boosting (XGB)[43]). Additionally, we incorporated a neural network-based feature selection method, where a fully connected feedforward network was trained on the dataset, and SHapley Additive exPlanations (SHAP) values were used to determine the most informative baits[44,45]. All methods were compared using subsets containing 30–80 baits, across 10 random seeds.

We compared these methods using 15 metrics that assess how well each bait subset aligned with the full dataset. These metrics were grouped into categories reflecting statistical component similarity, biological component similarity, overall biological preservation, and clustering preservation (Supplementary Table 2). To ensure quantification of both average and worst-case performance, we captured both mean and minimum values for key metrics. This distinction is particularly important, as some compartments are inherently more difficult to reconstruct, and strong average performance does not necessarily mean all components are well preserved (Fig. 3a–j).

We specifically quantified Pearson correlation, Cosine similarity and Kullback–Leibler (KL) divergence each of which offers a distinct statistical perspective on component similarity. Pearson correlation quantifies the linear relationship between original and subset components by measuring how their values co-vary[46]. Cosine similarity evaluates the alignment of components as vectors in high-dimensional space, making it particularly useful for comparing interaction patterns regardless of magnitude[47]. KL divergence assesses how well the subset preserves the overall distributional structure of the original data by measuring differences in probability distributions[48].

Beyond component similarity comparisons, we calculated the Adjusted Rand Index (ARI), purity score, and Jaccard index of enriched GO terms, all of which depend on clustering preys using the NMF components. NMF ARI, and NMF purity score measure how well prey clustering assignments are retained between the original and subset datasets, while the GO Jaccard index evaluates the overlap in enriched GO terms, reflecting how well biological information is preserved[49–51].

As expected, all methods performed markedly better than random subset selection across all datasets and all evaluation metrics. GENBAIT consistently achieved the highest scores in most mean NMF metrics. Interestingly, GENBAIT demonstrated significantly higher performance on the set of minimum NMF metrics compared to other methods. This indicates GENBAIT preserves all components without disproportionately underrepresenting any localization. In contrast, while other methods performed well on average, they consistently had at least one component with notably low values, suggesting a tendency to miss certain cellular compartments. This pattern was observed across all datasets. Among the other selection methods, differences were relatively small, though a general trend emerged. Machine learning-based approaches, particularly random forest models, tended to perform best, followed by linear regression-based models. Statistical tests ranked lowest. These trends remained consistent across different component numbers and were reflected in our comparative analyses (Supplementary Fig. 2).

To evaluate bait selection methods independently of NMF clustering, we analyzed a set of NMF-independent metrics focusing on two main aspects. First, we assessed the percentage of original preys that remained assigned to baits after subset selection. Second, we quantified the percentage of retrieved GO terms, measuring how well biological annotations were preserved. As expected, all bait selection methods outperformed random selection across these metrics. Beyond direct prey retention and GO term similarity, we incorporated three alternative clustering-based metrics to compare the structural consistency of subset-derived maps with the full dataset. We first constructed k-nearest neighbor (KNN) graphs[52] and applied Leiden clustering[53], evaluating the preservation of the original clustering structure using ARI. Additionally, we applied Gaussian Mixture Model (GMM)[54] clustering in two forms: a hard clustering approach, assessed via ARI, and a soft clustering approach, where we measured Pearson correlations between corresponding components in the original and subset maps (Fig. 4a–e). Since these NMF-independent metrics do not rely on NMF clustering, prey localization assignments do not necessarily align with the original study annotations. Given that most feature selection methods are optimized for NMF components or localization patterns derived from NMF, these alternative clustering and retrieval-based metrics serve primarily as complementary validation rather than as primary benchmarks for bait selection. Across these metrics, Ridge regression achieved the highest prey retention and GO term retrieval percentage. Random forest performed best for Leiden ARI, while GENBAIT ranked highest in GMM ARI, demonstrating its ability to retain prey grouping across independent clustering approaches. GBM showed the highest correlation between original and subset structures in GMM soft clustering. However, differences between methods in these metrics were relatively minor, reinforcing their secondary role in evaluating bait selection. These trends remained consistent across different numbers of clusters (Supplementary Fig. 3).

Additionally, to evaluate whether bait selection methods impact the global structure of prey–prey interaction networks, we compared topological metrics, including average shortest path length, betweenness centrality, degree distribution, and graph density, between the subset and original networks. Most methods performed comparably and preserved these properties well, while random selection showed the greatest variability. GENBAIT best retained the degree distribution and performed competitively across other metrics. Overall, these results indicate that most bait selection methods successfully preserve the topological structure of the original network (Supplementary Fig. 4).

Comparison with heuristic bait selection approaches

To further evaluate GENBAIT's effectiveness at bait subset selection, we compared it to three heuristic bait selection strategies: a high-yield method and two manual methods. In the high-yield approach, baits were selected based on having the highest number of significant preys in the full dataset, without regard to compartment. In the first manual method, an expert manually selected well-established marker proteins for each compartment based on relevant literature and published cell biology studies. In the second manual approach, compartments were first sorted alphabetically, and for each compartment, baits were ordered by decreasing prey count. The expert then iteratively selected the top bait(s) per compartment to balance spatial coverage and prey yield. These strategies were evaluated using the Human Cell Map dataset with bait subset sizes of 40, 60, and 80 (Supplementary Table 3).

Comparing these heuristic approaches to our established machine learning and statistical methods, we found GENBAIT consistently outperformed heuristic approaches, which ranked lowest across all evaluation metrics (Supplementary Figs. 5 and 6). While expert-curated and high-yield baits selections ensured the inclusion of key compartment-specific baits, they failed to optimize for the overall dataset structure, leading to weaker proximal interactome capture. Notably, all data-driven feature selection methods, including statistical and machine learning-based approaches, outperformed these heuristics. These findings emphasize the limitations of simple selection strategies and highlight the need for optimization-driven approaches like GENBAIT to generate representative and biologically meaningful bait subsets.

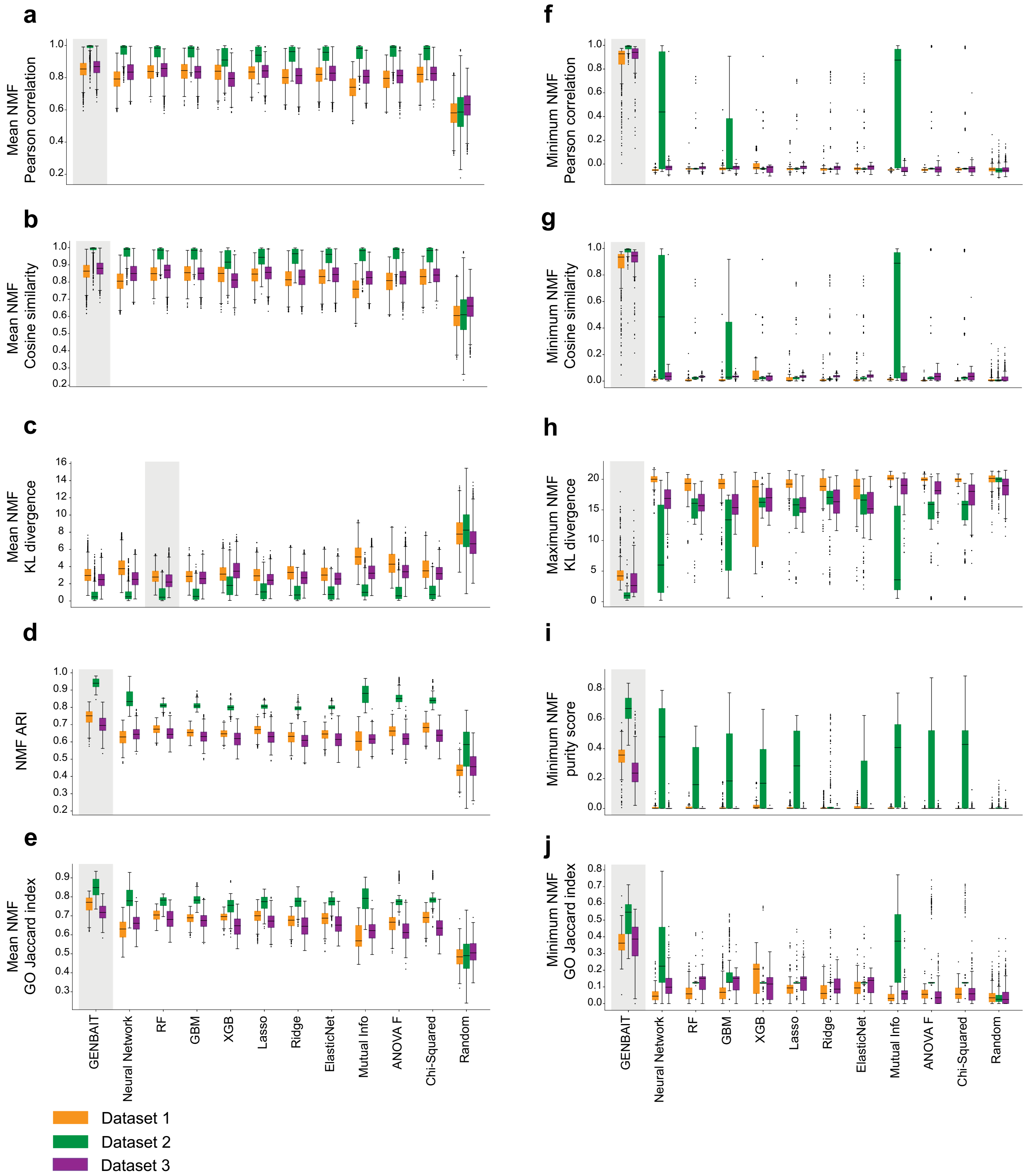

Fig. 3 | Benchmarking GENBAIT and other bait selection methods using NMF-derived metrics. Boxplots comparing GENBAIT, feature selection methods, and random selection based on mean scores: mean NMF Pearson correlation (**a**), mean NMF Cosine similarity (**b**), mean NMF KL divergence (**c**), NMF ARI (**d**), mean NMF GO Jaccard index (**e**); and min scores: min NMF Pearson correlation (**f**), min NMF Cosine similarity (**g**), max NMF KL divergence (**h**), min NMF purity score (**i**) and min NMF GO Jaccard index (**j**). Each method selected 30–80 baits over 10 random seeds. Box plots show median values; hinges are the 25th and 75th percentiles; whiskers indicate 1.5× the interquartile range (IQR). Highest average median values have been shaded in gray.

Assessing the applicability of selected baits across different cell lines

To assess whether selected baits are ubiquitously expressed, we extracted ProteomicsDB[55] expression data for the top baits selected by GENBAIT from the Human Cell Map across an extended set of 11 cell lines (see "Methods"). Of the 46 baits that could be analyzed, 71.7% (33/46) were detectably expressed in all 11 cell lines, and 100% (46/46) were expressed in at least half of them. This widespread expression indicates that the selected baits are not overly optimized for a single cell line (Supplementary Fig. 7). Additionally, to assess whether

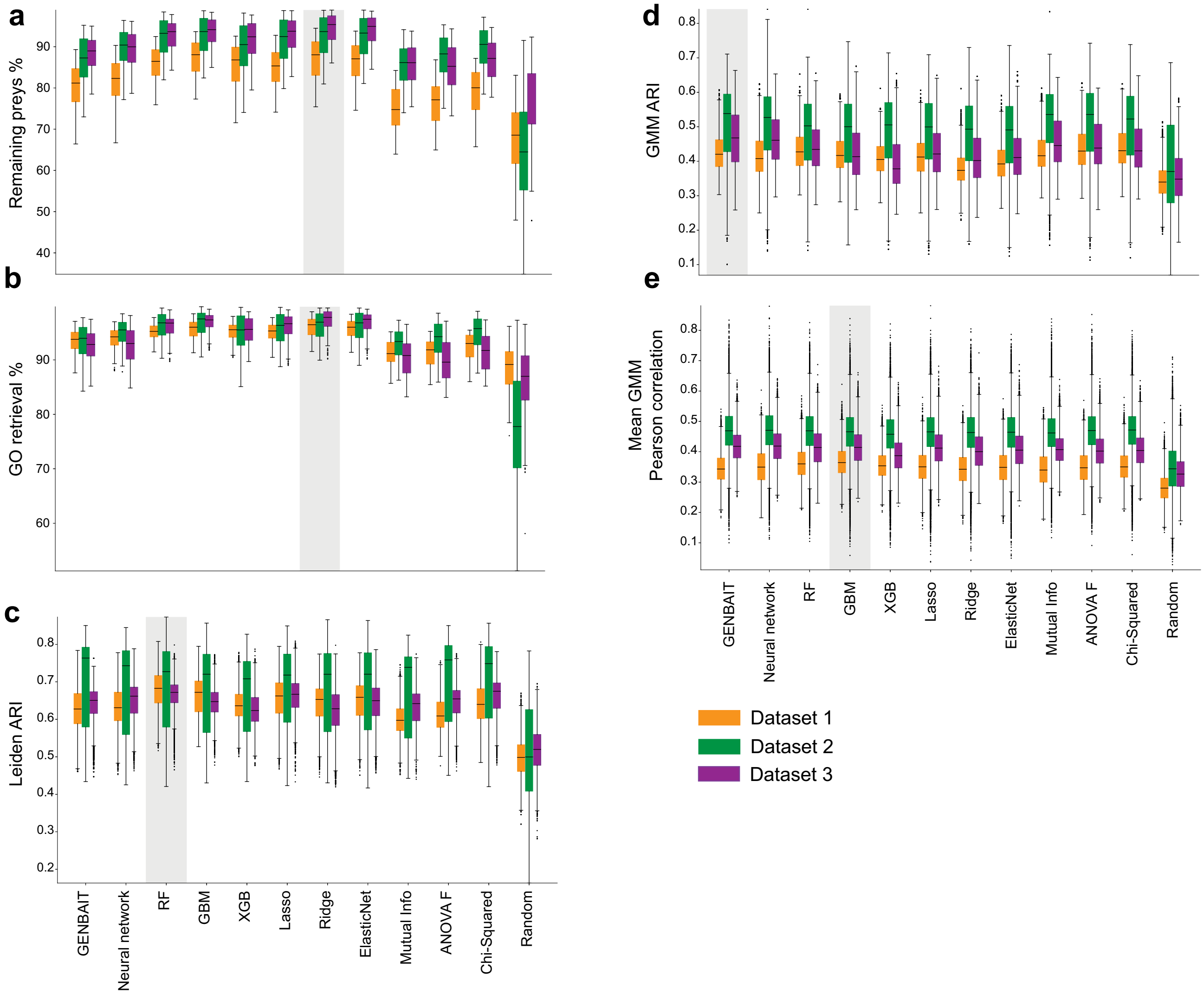

Fig. 4 | Benchmarking GENBAIT and other bait selection methods using non NMF-derived metrics. a–e Boxplots comparing GENBAIT, benchmarking methods, and random selection based on non-NMF metrics, including the remaining preys percentage (**a**) and GO retrieval percentage (**b**); Leiden clustering of KNN graphs at three resolutions (**c**); and GMM hard clustering (**d**) and soft clustering (**e**) at four cluster numbers. Each method selected 30–80 baits over 10 random seeds. Box plots show median values; hinges are the 25th and 75th percentiles; whiskers indicate 1.5× the interquartile range (IQR). Highest average median values have been shaded in gray.

GENBAIT-selected baits are influenced by cell cycle variation, we examined their classification in the Human Protein Atlas (HPA)[3]. Of the 50 baits, 12 were not found in the HPA database. Among the remaining 38, only one (DCTN1) was annotated as cell cycle dependent.

To further evaluate bait selection beyond HEK-293, we simulated prey–bait matrices based on prey–bait expression ratios between HEK-293 and different cell lines (see "Methods"). This allowed us to model how expression differences might impact reconstruction of proximity interactome using HEK-293-derived baits. We then assessed the performance of GENBAIT and other bait selection methods within these simulated datasets using metrics we previously defined. Despite expression variability, HEK-293-selected baits remained effective in reconstructing proximity interactomes across different cell lines, consistently achieving high scores across our evaluation metrics and outperforming random selection (Supplementary Note 1). While some compartments exhibited lower correlation values due to missing preys or baits, the overall interactome structure was largely preserved. We further confirmed that GENBAIT performance is not simply driven by global prey expression similarity: mean NMF Pearson scores and overall prey expression correlation with HEK-293 were not correlated ($r = -0.27$, $p = 0.452$; Supplementary Fig. 8). This suggests that HEK-293-derived bait subsets provide a reliable foundation for spatial proteomic mapping across diverse cellular environments.

Robustness of subcellular map reconstruction at different bait subset sizes and random seeds

To assess the stability of bait subset selection methods, we evaluated their performance across different bait subset sizes, ranging from 30 to 80 baits. For each method, we generated 10 independent subsets per bait length to measure consistency across different random initializations. In GENBAIT, this randomness comes from the initial bait subset, which is selected randomly. In other methods, the randomness arises from the random train-test split of data, which differs with each seed. We then computed the mean NMF Pearson correlation scores to quantify how well each subset preserved the organization of the original dataset (Fig. 5a). Across all methods and datasets, the mean NMF Pearson correlation score increased with the number of selected baits, demonstrating that larger subsets generally retain more information from the full dataset. GENBAIT consistently achieved the highest scores across all bait lengths and exhibited a gradual, steady increase

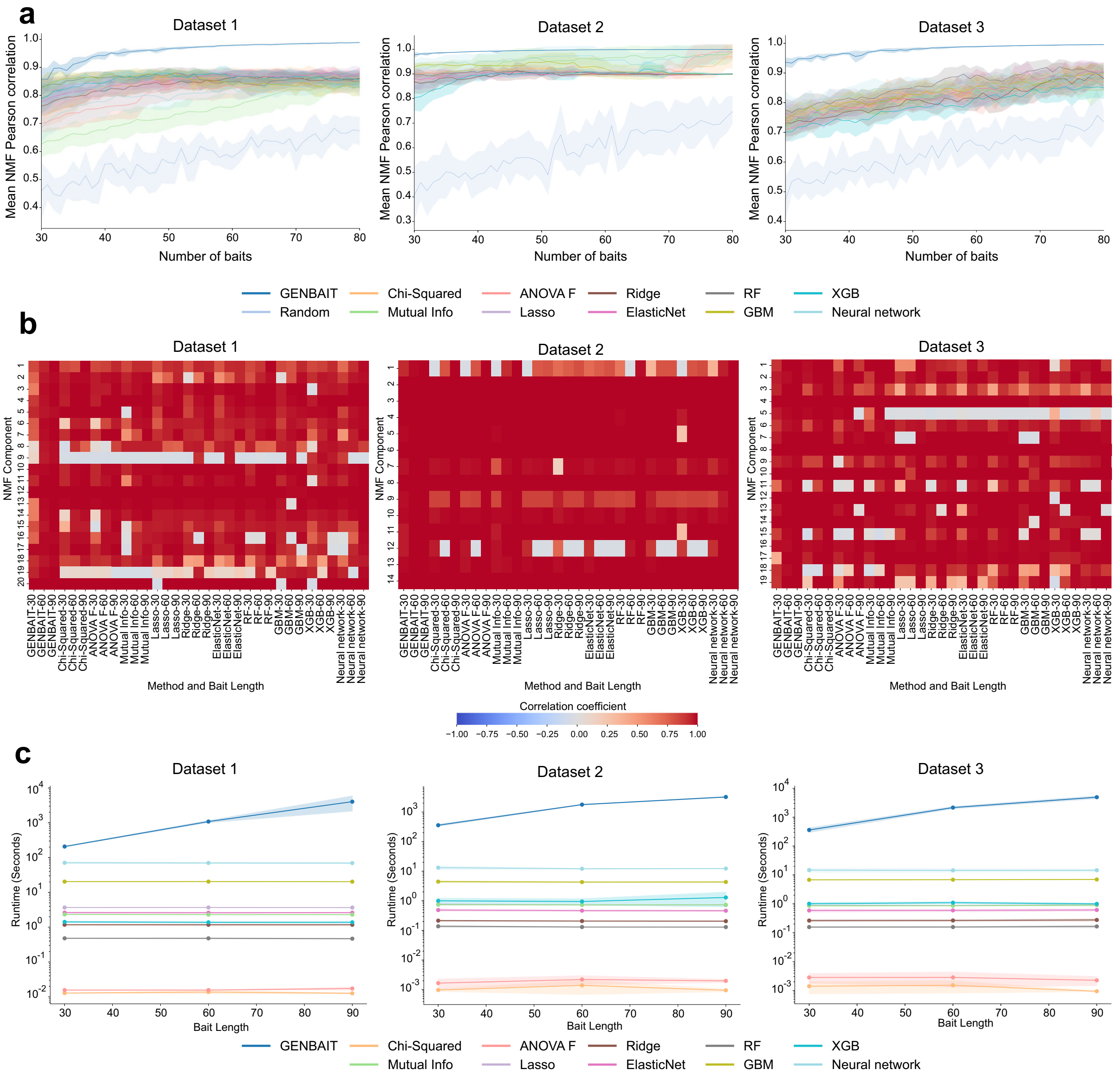

Fig. 5 | Impacts of varying the subset size and random initializing seed on the performance of bait subset selection methods. a Mean NMF Pearson correlation scores for each bait selection method across different bait subset sizes. Colored lines indicate the mean across random seeds, and shaded regions represent ±1 standard deviation. **b** Heatmap showing the correlation of individual NMF components between the original dataset and subsets generated by different bait selection methods across various bait subset sizes. Each column represents a method at a specific bait length, while each row corresponds to an NMF component. **c** Runtime analysis of bait selection methods across different bait subset sizes. Lines represent the mean runtime (log scale) over three independent runs, and shaded regions indicate ±1 standard deviation.

in performance, ultimately reaching a plateau. This smooth progression without major fluctuations suggests that GENBAIT optimizes bait selection without being trapped in local optima. In contrast, other methods displayed substantial fluctuations across different bait sizes, and their performance did not stabilize as clearly as GENBAIT. This inconsistency suggests that these methods may be more sensitive to the initial random selection of baits, leading to variable results. Notably, GENBAIT exhibited minimal variance, demonstrating its robustness to initialization effects. In contrast, all other methods showed remarkably larger variations, highlighting their sensitivity to random initialization and the potential instability of their selected subsets. Together, these results emphasize that while all methods improve with increasing bait numbers, GENBAIT not only outperforms other approaches but also provides the most stable and reproducible bait selection.

To determine whether certain NMF components consistently require more baits for accurate reconstruction, we analyzed component-wise Pearson correlation values across bait subset sizes. This helped assess whether specific subcellular compartments are inherently harder to capture due to biological complexity, spatial organization, or technical constraints. Some components exhibited persistently low correlation values across all methods, particularly at smaller bait sizes (Fig. 5b). In dataset 1, these included the cytoskeleton (component 8), centrosome (component 9), and cytoplasmic

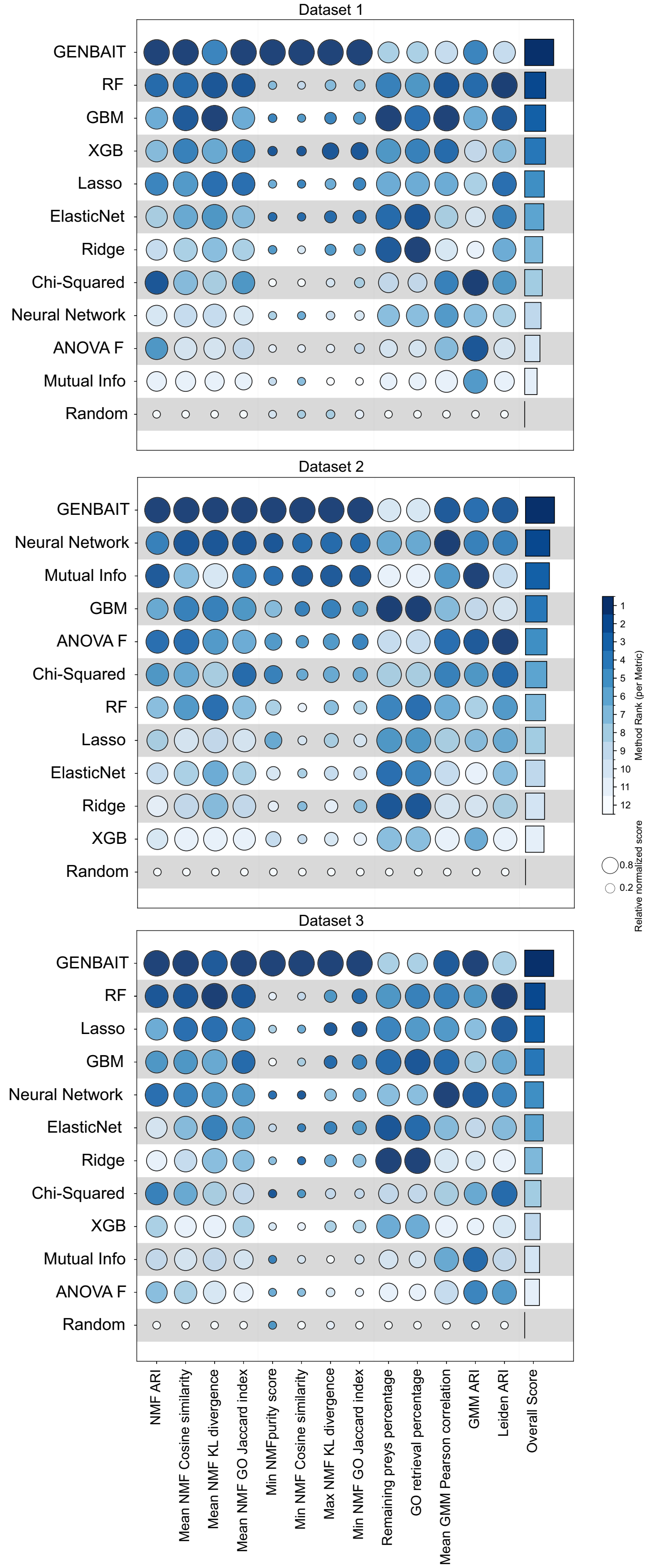

Fig. 6 | Overall scoring and ranking of bait selection methods. A heatmap of the scores of various bait selection methods across multiple metrics. The color of each circle represents its ranked normalized score. Size of each dot indicates its normalized score in the column. The overall scores are normalized average scores for each method over all metrics.

ribonucleoprotein granules (component 19). In dataset 2, cytoplasmic ribonucleoprotein granules (component 1) and cytoplasmic stress granules (component 12) were the most challenging, while in dataset 3, splicing machinery (component 5), PML bodies (component 15), and polycomb complexes (component 18) were the hardest to recapitulate. Notably, many of these difficult-to-reconstruct components correspond to biologically complex compartments that encompass diverse sets of interacting proteins. Our results suggest that the difficulty in reconstructing these components is at least partly driven by their size and heterogeneity, which demand broader bait coverage to capture their full interactions. GENBAIT consistently improved correlation values for these challenging components as bait number increased, indicating that it effectively prioritizes baits critical for preserving complex subcellular structures (Fig. 5b). In contrast, other methods often showed uneven gains, with some components improving at the cost of others. The ability of GENBAIT to progressively recover low-correlation components without sacrificing the rest of the proteome structure highlights its strength in maintaining spatial organization across compartments of varying size and complexity.

Runtime analysis

To evaluate computational efficiency, we analyzed runtime at bait subset sizes of 30, 60, and 90, repeating each run three times (Fig. 5c). Most methods exhibited a consistent runtime regardless of bait size, as they primarily rely on ranking all baits. In contrast, GENBAIT's runtime increased with bait size due to its iterative optimization process. Neural network-based selection took longer than other ML-based methods but was still much faster than GENBAIT. Despite its higher computational cost, GENBAIT remains feasible, requiring under two hours to select 90 baits in a dataset of ~200 baits. While the most time-intensive, a runtime of under two hours is minimal compared to the duration of a typical proximity labeling experiment, which often spans weeks to months.

Recommendations for bait subset selection for BioID

To facilitate a comparison of GENBAIT and other feature selection methods, we generated overall scores and ranked them (Fig. 6). To do so, we computed average scores for each evaluation metric and scaled them such that the minimum value was 0 and the maximum was 1, ensuring consistency across different metrics. The overall scores for each method were then determined by averaging these scaled scores across all metrics, except for the primary metrics, which are the mean and minimum NMF Pearson correlation score, that are directly optimized by GENBAIT. Finally, these overall scores were also scaled to a 0–1 range for consistency.

GENBAIT ranked highest across all datasets, consistently outperforming other methods. While some bait selection methods performed well in specific datasets, their performance was inconsistent across different datasets and metrics. This variability highlights the limitations of purely statistical or machine learning-based approaches, which may be biased toward the most dominant features rather than ensuring the comprehensive retention of all subcellular components. Notably, clustering-based metrics tended to favor methods that prioritize identifying the most dominant localization of proteins. However, our main objective is to preserve all localizations within the subsets, which these approaches may not fully capture. GENBAIT, on the other hand, performed best in NMF-dependent metrics, demonstrating its ability to optimize bait selection for reconstructing subcellular localization maps. It effectively retains information across multiple localizations, making it a suitable choice for protein multi-localization studies.

Overall, no single method was universally optimal, but feature selection consistently outperformed random bait selection. Our

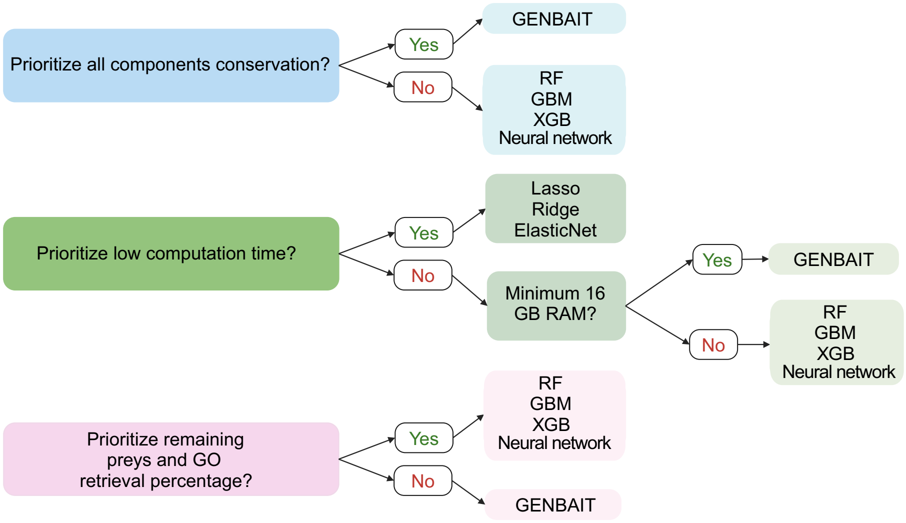

Fig. 7 | Guidelines for choosing a bait selection method. The optimal method will depend on the question asked, the experimental priorities, and the available computational resources.

recommendation framework (Fig. 7) provides guidance to allow users to tailor choices based on specific requirements. When adequate system memory (RAM) is available, GENBAIT is the clear choice for preserving clustering structures and optimizing bait selection. For faster alternatives, regression-based methods like lasso, ridge, or elastic net offer a balance between speed and accuracy. When efficiency is a priority, ensemble methods such as random forest, GBM, XGB, and neural networks provide a practical option, performing well across NMF-based and biological retrieval metrics.

Discussion

Our study demonstrates the potential of feature selection methods, including a genetic algorithm-based approach, GENBAIT, in addressing the challenge of selecting optimal bait subsets for BioID experiments. By developing and applying a comprehensive evaluation pipeline with 15 diverse metrics, we provide a robust framework for assessing and comparing these methods in the context of proximity proteomics research. Given the high cost and complexity of large-scale BioID experiments, optimizing bait selection is essential for balancing experimental efficiency with biological coverage. To address this challenge, we systematically evaluated feature selection methods to determine their effectiveness in preserving spatial proteomic organization while minimizing the number of required baits.

A key contribution of this work is demonstrating that established feature selection methods can be effectively adapted for BioID bait selection, significantly outperforming random selection. Additionally, our introduction of a set of benchmarking metrics provides a standardized approach for evaluating bait subsets, allowing researchers to make data-driven decisions tailored to their specific research goals. These metrics serve as practical tools for selecting baits that maintain the structural and functional integrity of the original dataset, ensuring that biologically relevant interactions are preserved. GENBAIT showed strong performance across multiple metrics, particularly those derived from NMF-based analyses. However, it did not universally outperform all other methods across every metric, highlighting the importance of selecting a feature selection strategy that aligns with the specific objectives of a study rather than relying on a single approach. Machine learning and statistical methods also performed well, particularly for metrics that emphasize primary localization rather than multi-localization. This underscores the need to consider both the method's strengths and the biological context when choosing an optimal bait selection strategy. While not strictly statistically independent, the 15 metrics in our benchmarking framework are complementary as evidenced by the fact no method consistently outperforms others. Consequently, collapsing them into one overall rank can mask important differences between methods. In practice, we advise users to interpret the results hierarchically: NMF-based metrics quantify how well structural information is preserved, while biological metrics assess functional coverage. Depending on the study focus, these metrics can be weighted differently.

While our study focuses on BioID-based proximity proteomics, the underlying framework of GENBAIT is not limited to a specific labeling enzyme. Beyond BioID, proximity labeling methods such as APEX generate similar bait–prey interaction matrices, suggesting that the same feature selection principles can be applied to optimize bait selection across multiple proximity proteomics approaches.

Although GENBAIT was originally developed for large-scale proximity proteomics studies, such as mapping organelles and spatial interaction networks, it can also be useful to support more targeted applications, using source data (e.g., the Human Cell Map dataset). For example, if a researcher is interested in just a few compartments, they can focus on that specific region of the proximity map and apply the full GENBAIT workflow to select the most informative baits within that subset. This makes GENBAIT a flexible tool that can support both system-wide studies and more focused, targeted experiments. Although GENBAIT requires an initial dataset, this upfront investment can substantially reduce the number of subsequent experiments by focusing efforts on the most informative baits. For smaller-scale projects, using the growing list of large reference maps minimizes the need for extensive pre-screening while still providing the benefits of reduced experimental costs and improved coverage.

Despite its advantages, GENBAIT has certain limitations. First, although our computational approach provides valuable insights, we did not experimentally validate the bait subsets generated by GENBAIT or other methods. Additionally, while GENBAIT's optimization-based approach ensures high-quality bait selection, it is computationally more demanding than other methods. Although runtime remains feasible, it may be a limiting factor for researchers with constrained computational resources. Future iterations of GENBAIT could focus on improving efficiency through parallelization and algorithmic refinements without sacrificing performance.

In conclusion, our study highlights the utility of feature selection methods for BioID bait subset selection and provides a comprehensive framework for evaluating these methods. While GENBAIT offers a highly competitive approach, particularly for studies focused on multilocalization, it is not a one-size-fits-all solution. Researchers should carefully consider their specific research objectives, available resources, and the trade-offs between accuracy and computational efficiency when selecting a feature selection method. By providing a structured framework for bait selection and formalizing benchmarking metrics, this study lays the foundation for optimizing proximity labeling experiments at scale.

Methods

Data preprocessing

The datasets were preprocessed using a custom pipeline to ensure data quality and consistency. First, the average spectral counts for each prey protein identified in the negative controls were subtracted from its average spectral counts observed for a given bait, giving a corrected average. Preys were then filtered to include only high-confidence interactions (i.e., those with a Bayesian false discovery rate (BFDR)[18] ≤ 0.01 for datasets 1 and 3, and SaintScore ≥ 0.95 for dataset 2). These data were pivoted to create a matrix with the baits in rows and the preys in columns, which was filled with control-subtracted average spectral count values. The MinMaxScaler was applied to scale the data, ensuring that feature values were normalized between 0 and 1.

Benchmarking methods

Data preparation. Each preprocessed dataset was randomly split into 80% for training and 20% for testing, ensuring reproducibility with a fixed seed. To prevent data leakage, non-negative least squares (NNLS)[56] fitting was applied during decomposition. NMF was first applied on the full dataset to generate scores and basis matrices while preserving the original number of components. The basis matrix was then used to assign each prey to the component with its highest score, forming the target variables for feature selection. NMF was separately applied to the training subset to obtain its scores and basis matrices. For the testing subset, the basis matrix was initialized with zeros and computed using NNLS to maintain consistency with the training decomposition. This ensured that information from the test set did not influence the training process. The resulting matrices were combined to create a complete set for training and testing, where transposed data matrices served as input variables, and target labels were based on the assigned NMF components.

Feature selection methods. We used multiple feature selection methods to identify the most informative features:

- Chi-squared tests[57] (chi2 in scikit-learn[58] python package) were used to assess the dependency between each feature and the assigned class label, which was determined based on the component with the highest value following NMF. The test evaluates whether the distribution of a feature's values differs significantly across classes by comparing the observed values to what would be expected if there were no relationship between the feature and the class label. The chi-squared score measures how strongly a feature is associated with a class label. A higher score indicates that a feature varies significantly across different NMF components, meaning it is more informative for classification. All baits are ranked based on their chi-squared scores, and for each desired bait length, the top-ranked features are selected.
- ANOVA F tests[59] (f_classif in scikit-learn python package) were used to identify features that show significant differences across the assigned class labels, which were determined based on the highest NMF component values. This test evaluates whether the mean values of a feature differ significantly between classes by comparing the variance within each class to the variance between classes. The F-score quantifies how much a feature's values vary across different NMF components. A higher F-score indicates that a feature shows greater variation between classes than within them, making it more informative for classification. All features are ranked based on their F-scores, and for each desired bait length, the top-ranked features are selected.
- Mutual information[60] (mutual_info_classif in scikit-learn[58] python package) was used to measure the dependency between each feature and the assigned class label, which was determined based on the highest NMF component values. Unlike statistical tests that assume a linear relationship, mutual information captures both linear and nonlinear associations by quantifying how much knowing the value of a feature reduces uncertainty about the class label. A higher mutual information score indicates that a feature provides more information about the assigned class, making it more relevant for selection. All features are ranked based on their mutual information scores, and for each desired bait length, the top-ranked features are selected.
- Lasso regression[61] (Logistic Regression in scikit-learn python package) was used to perform feature selection by applying an L1 penalty to a logistic regression model with saga[62] solver. This penalty encourages sparsity in the model by driving the coefficients of less important features to exactly zero, effectively removing them from consideration. After training the model, the absolute values of the learned coefficients were examined. Features with larger absolute coefficients were considered more important, as they contributed more to distinguishing between classes based on the NMF component assignments. All features were ranked based on these absolute coefficient values, and for each desired bait length, the top-ranked features were selected.
- Ridge regression[61] (Logistic Regression in scikit-learn python package) was used for feature selection by applying an L2 penalty with saga solver, which discourages large coefficient values but does not force them to zero. Unlike Lasso, which performs strict feature selection by eliminating some coefficients, Ridge retains all features while reducing the impact of less informative ones. After training the model, the absolute values of the learned coefficients were computed to rank all baits. Features with the highest absolute coefficients were selected, ensuring that the most important predictors were chosen for each desired bait length.
- Elastic net regression[63] (Logistic Regression in scikit-learn python package) was used for feature selection by combining both L1 and L2 penalties, balancing sparsity and regularization. This method benefits from Lasso's ability to shrink some coefficients to zero while leveraging Ridge's stability in handling correlated features. After training the model, the absolute values of the learned coefficients were computed to rank all baits. Features with the highest absolute coefficients were selected for each desired bait length.
- Random forest[64] (RandomForestClassifier in scikit-learn python package) selects important features by measuring how much they help separate different classes in decision trees. Each tree in the model is trained on a random part of the data, and at each split, the algorithm picks the feature that best groups similar samples together. Features that consistently improve grouping are considered more important. The importance scores from all trees are averaged, and features are ranked based on these scores. For each bait length, the top-ranked features are selected.
- GBM[65] (GradientBoostingClassifier in scikit-learn python package) build a series of decision trees, where each tree focuses on correcting the mistakes of the previous one. The importance of each feature is determined by how much it improves the model's predictions at each split. Features that contribute more to reducing errors are assigned higher importance scores. After training, features are ranked based on these scores, and the top-ranked features are selected for each bait length.
- XGB[66] (XGBClassifier in XGBoost[43] python package)is an optimized version of gradient boosting that uses efficient techniques

like regularization and handling missing values to improve performance. It measures feature importance based on how often a feature is used in tree splits and how much it helps reduce errors. Features with higher scores are ranked higher, and for each bait length, the top-ranked features are selected.

- Neural network-based feature selection was performed using a fully connected feedforward neural network trained with PyTorch Lightning[67]. The model consisted of an input layer, a hidden layer with ReLU activation, and an output layer corresponding to the number of components. The network was optimized using the Adam optimizer[68] and trained with cross-entropy loss. Once trained, SHAP values were computed to estimate the contribution of each feature to the network's predictions. The SHAP values were averaged across all samples, and the features with the highest SHAP importance scores were selected for each bait length.

Bait subset selection with GENBAIT

Algorithm initialization. An initial population of solutions was generated, with each individual represented as a binary vector. The length of the vector corresponded to the number of baits in the dataset, with each element indicating the presence (1) or absence (0) of a particular bait. A predetermined random seed was used to initialize the population, ensuring the reproducibility of the results.

Fitness function. The fitness of each individual was determined using a custom fitness function, evalSubsetCorrelation, which assessed the subset's representativeness of the original dataset based on the correlations between their corresponding NMF components. Subsets outside the pre-specified size range were heavily penalized to enforce size constraints. For valid subsets, NMF was applied to extract basis matrices, followed by alignment using the Hungarian algorithm[36] to ensure that the components were ordered as in the original dataset. The fitness score was calculated using the correlation matrix diagonal values, with penalties for negative correlations to discourage unrepresentative feature combinations.

The fitness function, $f(I)$, for an individual I in the genetic algorithm is defined as follows:

$$f(I) = \begin{cases} 0 & \text{if} |\mathrm{S}(I)| \notin \text{subset range} \\ \text{mean}\left(\text{diag}\left(\text{Corr}\left(B, B^{*}\right)\right)\right) - \text{Penalty}(I) & \text{otherwise} \end{cases} \tag{1}$$

Where:

- $|\mathrm{S}(I)|$ is the size of the feature subset represented by the individual I.
- Subset range is the allowable size range for the feature subset.
- B is the basis matrix from NMF applied to the original dataset.
- B* is the reordered basis matrix from NMF applied to the subset of data corresponding to I.
- Corr(B, B*) calculates the correlation matrix between column of matrices B and B*.
- diag(**X**) extracts the diagonal elements of matrix X.
- Penalty(I) is a function that applies a penalty based on the number of negative values in the diagonal of the correlation matrix, calculated as:

$$\begin{aligned} \text{Penalty}(I) &= \text{Penalty factor} \\ &\times (\text{number of negative values in diag}(\text{Corr}(B, B^{*}))) \end{aligned} \tag{2}$$

- Penalty factor is a predefined constant.

Genetic operators. Crossover and mutation were implemented to generate new solutions[69]. Crossover was performed using a two-point crossover method (cxTwoPoint) with a specified probability. Mutation (mutFlipBit) involved flipping bits in the individual's binary representation with a mutation probability. Individuals were selected for the next generation based on fitness, using a tournament selection method (selTournament).

Algorithm execution. GENBAIT ran for a defined number of generations (n_generations), each involving selection, crossover, and mutation. The algorithm's progress and population dynamics were tracked using a logbook, and the best-performing individuals were recorded in a hall-of-fame.

Computational environment and resources. The GENBAIT algorithm was implemented in Python, using the DEAP[70], NumPy[71], Pandas[72] libraries for evolutionary computations, numerical operations, and data processing, respectively. NMF decomposition was performed using the scikit-learn library. Nonnegative Double Singular Value Decomposition was used for NMF initialization, and the regularization parameter (l1_ratio) was set to 1. The Hungarian algorithm was applied through the SciPy[73] library's linear_sum_assignment function.

Algorithm parameters. Specific parameters used in GENBAIT, including population sizes, crossover and mutation probabilities, and the number of generations, were chosen based on preliminary experiments to balance computational efficiency and the quality of the feature selection process (see Supplementary Table 4).

Random bait subset selection

To establish a baseline for comparison, we generated 1000 random subsets of features by randomly selecting a set of indices from the original dataset, with the subset size falling within a specified range between 30 and 80. We then evaluated their utility as a reference point against which the performance of more systematic feature selection methods could be assessed, based on corresponding NMF components correlations.

Validation metrics

Mean and minimum NMF Cosine similarity. To assess how well bait subsets retained the structure of the full dataset, we computed the mean and minimum Cosine similarity after aligning the NMF basis matrices. Subsets generated by random selection, GENBAIT, and other methods were subjected to NMF, and their basis matrices were aligned with the original dataset using the Hungarian algorithm by minimizing the overall dissimilarity. After reordering the subset components, Cosine similarity was calculated. Mean Cosine similarity measured overall similarity between the full and subset datasets, reflecting global structural consistency. Minimum Cosine similarity captured the weakest-matching component, identifying cases where certain subcellular localizations were not well preserved.

Mean and maximum NMF Kullback–Leibler (KL) divergence. To assess how well the selected bait subsets preserved the probabilistic distribution of prey assignments within each NMF component, we computed the mean and maximum KL divergence. Unlike Cosine similarity, which measures structural alignment, KL divergence quantifies how much the probability distribution of the subset deviates from the original dataset. After performing NMF on both the full and subset datasets, we aligned the subset basis matrix with the original using the Hungarian algorithm. To ensure valid probability distributions, each basis matrix was normalized by summing component values to one, with a small epsilon added to prevent division errors. KL divergence was then calculated between corresponding components, measuring the relative information loss in prey distributions. Mean KL divergence provided an overall measure of how much, on average, the subset components deviated from the original distributions. A lower mean value indicated that the subset retained the overall localization structure well. Maximum KL divergence highlighted the most divergent component, identifying cases where a specific subcellular localization was poorly preserved.

NMF ARI and min NMF purity score. To evaluate how well the selected bait subsets preserved the clustering assignment of the original dataset, we computed NMF ARI and the minimum NMF purity score. Subsets generated by different methods were subjected to NMF, and their basis matrices were aligned with the original dataset using the Hungarian algorithm. ARI quantifies the agreement between component assignments of preys in the original and subset datasets while correcting for chance. A value of 1 indicates perfect clustering retention, whereas 0 represents random clustering. To calculate ARI, preys were assigned to the component with the highest value in both the original and subset basis matrices, and clustering similarity was measured. Minimum NMF purity score evaluates how well individual components retained their original composition. It measures the fraction of preys within each component that remained consistently assigned to the same cluster after subset selection. A high purity score indicates that most preys in a component were preserved, while a low score suggests that some compartments were misclassified. While ARI provides a global measure of clustering similarity, the minimum purity score ensures that no single component is significantly disrupted.

Mean and minimum NMF GO Jaccard index. To evaluate how well each bait subset preserved the biological relevance of the original dataset, we computed the mean and minimum GO Jaccard index. Bait subsets generated through feature selection methods underwent NMF, and their basis matrices were aligned with the original dataset using the Hungarian algorithm. The GO Jaccard index quantifies the overlap between GO terms enriched in the NMF components of the original and subset datasets. It is calculated as the size of the intersection divided by the union of GO terms for each component, with higher values indicating greater biological consistency. GO term enrichment was performed using g:Profiler, retrieving the most significant GO:CC terms associated with each NMF component in both the original and subset datasets. The mean GO Jaccard index represents the overall functional similarity across all components, providing a general measure of how well the subset retains biological annotations from the full dataset. The minimum GO Jaccard index, in contrast, highlights the weakest-matching component, identifying cases where specific subcellular localizations or functional groups are disproportionately affected.

Leiden clustering. To evaluate how well the selected bait subsets preserved the neighborhood structure of the original dataset, we performed Leiden clustering using leidenalg[53,74] python package on KNN graphs constructed from the original and subset datasets and computed the ARI between their cluster memberships. In more details, we constructed a KNN graph using igraph[75] python package, where vertices represented preys and edges indicated neighbor relationships. The number of neighbors (k) was set to 20, and connectivity-based graphing was used. The resulting graph was then converted into a network representation for clustering analysis. Leiden clustering was applied to the KNN graph using a community detection algorithm optimized for modularity. The clustering was performed at different resolution parameters (0.5, 1, 1.5) to explore the robustness of cluster structures. Each bait was assigned to a cluster based on its neighborhood relationships, and cluster memberships were recorded. Clustering was performed separately for the original dataset and for all generated subsets. For each subset, a new KNN graph was constructed and clustered using the same parameters as for the original data. The similarity between the cluster memberships in the original and subset datasets was quantified using the ARI, providing a measure of cluster preservation. The entire process was repeated across 10 random seeds to assess consistency.

GMM hard clustering. We applied GMM clustering using scikit-learn python package to both the original and subset datasets and quantified the similarity of their cluster assignments using ARI, to evaluate how well the selected bait subsets preserved the prey cluster structure of the original dataset. For each bait subset, GMM was applied to assign preys to clusters, with models trained using 15, 20, 25, and 30 clusters for datasets 1 and 3, and 5, 10, 15, and 20 clusters for dataset 2. Only preys present in both the original and subset datasets were considered in the evaluation. The bait subset was clustered separately, and its assignments were aligned with those from the full dataset. ARI was computed to quantify the similarity between the original and subset cluster assignments, ensuring that the subset preserved the structural organization of the full dataset. The entire process was repeated across 10 random seeds to assess consistency.

GMM soft clustering. To evaluate how well the selected bait subsets preserved the probabilistic prey cluster structure of the original dataset, we applied soft GMM clustering to both the original and subset datasets and quantified the similarity of their cluster probability distributions using mean diagonal correlation. GMM soft clustering was applied using 15, 20, 25, and 30 clusters for datasets 1 and 3, and 5, 10, 15, and 20 clusters for dataset 2. Instead of assigning each prey to a single cluster, this approach estimated the probability of each prey belonging to multiple clusters. For reference, the original dataset was first clustered, and the probability distributions of preys across clusters were stored. The same clustering procedure was then applied to each bait subset, generating probability distributions that were compared to those from the full dataset. The alignment between original and subset clusters was optimized using the Hungarian algorithm, and the correlation between corresponding cluster probability distributions was computed as a measure of similarity. The mean diagonal correlation of the reordered probability matrix was used to quantify how well the subset preserved the probabilistic structure of the original dataset. Higher values indicated greater retention of prey localization patterns, while lower values suggested a loss of structural information. The entire process was repeated across 10 random seeds to assess consistency.

Remaining preys percentage. We next quantified how well different feature selection methods preserved relevant biological information in the original datasets by calculating remaining preys percentage. For each subset generated, we calculated the proportion of non-zero preys, which means preys that have at least one interaction with one of the selected baits, to assess the retention of significant preys.

GO retrieval percentage. We then analyzed how well bait subsets retain biological annotations by measuring the percentage of GO terms retrieved from the original dataset. The analysis began by loading the gene annotation file (GAF)[76,77] data into a structured format. Each entry in the GAF file, adhering to the GAF 2.1 specification, was parsed to extract required information, including the database identifier, object symbol, and GO ID.

Each subset's genes were mapped to GO:CC terms, with a focus on identifying and retaining terms within a specified maximum term size (=1000) to mitigate the influence of broader terms focusing on a certain level of specificity. We then quantified the overlap in GO terms between the original dataset and the subsets. This was achieved by calculating the percentage of common GO terms, providing a measure of preservation of biological relevance across different feature selection methods.

Statistical analysis. We assessed the statistical significance of differences between feature selection methods using two-sided Mann–Whitney U test, a non-parametric test that compares the distributions of scores between methods. Pairwise comparisons were performed across all methods, and p values were adjusted for multiple testing using the Benjamini–Hochberg correction to control the false discovery rate.

Benchmarking the methods using different metrics and calculating overall scores. To compare GENBAIT's performance with that of other feature selection methods, we calculated average scores for each evaluation metric, which were normalized to 0–1 to ensure the comparability of different metrics. Overall scores for each method were calculated by averaging the normalized scores across all metrics, excluding the main metrics (mean and minimum NMF Pearson correlation scores). This approach allowed us to objectively assess and rank the performance of each method across multiple evaluation criteria.

Network topology analysis. To evaluate how different feature selection methods affect the structural properties of prey–prey interaction networks, we computed network topology metrics for each method's selected subset. A prey–prey adjacency matrix was constructed, generating an undirected graph representation of the dataset using networkx[78] Python package. We then calculated multiple graph topological properties to assess how well each method preserved the network structure.

Average shortest path length is the average number of steps required to travel between two nodes. A shorter path length suggests that the network remains well-connected, while longer paths indicate that interactions have become more dispersed due to bait selection. Betweenness centrality quantifies the extent to which a node acts as a bridge in the network by measuring the number of shortest paths that pass through it. High betweenness values indicate proteins that facilitate communication between different regions of the network. If a subset network has significantly lower betweenness values, it suggests that some key bridging interactions may have been lost. Degree distribution measures the average number of connections per prey. A high degree suggests that certain proteins serve as key interaction hubs, while a lower degree indicates a more fragmented network. Comparing the degree distributions of the subset and original networks helps determine whether bait selection disproportionately removes highly connected preys. Network density represents the proportion of possible edges that are present in the network. A high density means that preys are highly interconnected, while a lower density indicates sparser interactions. This metric helps evaluate whether the selected baits lead to a network that remains as interconnected as the original.

To compare the networks derived from different bait selection methods, we calculated the ratio of each metric between the subset and the original network. A ratio close to 1 suggests that the method effectively maintains the original network's structural properties, while deviations indicate changes in connectivity patterns. The metrics were evaluated across bait lengths ranging from 30 to 80 and 10 random seeds to ensure robustness.

Heuristic bait selection

We implemented three heuristic bait selection strategies for comparison with GENBAIT and other computational methods. In the first approach, baits were selected solely based on the highest number of significant preys (SAINT BFDR ≤ 0.01) in the full dataset, regardless of their subcellular localization. In the second approach, an expert manually reviewed all compartments using Human Cell Map annotations and identified well-established marker proteins for each compartment based on relevant literature and published cell biology studies. For each compartment, the overlap between these known markers and the available baits in the Human Cell Map was determined, and matching baits were selected. In the third approach, an expert curated bait panels by iterating through all compartments (based on Human Cell Map annotations), sorted alphabetically and by decreasing prey count, and then selecting the top-ranking baits per compartment to balance biological diversity and prey coverage. Multi-localized baits were expanded across compartments where applicable, and rounding adjustments were applied to ensure the bait panel matched the desired size. All three strategies were used to generate bait subsets of size 40, 60, and 80 for benchmarking.

Bait expression across cell lines

To evaluate whether GENBAIT-selected baits maintain consistent expression across diverse cellular contexts, we analyzed protein expression profiles using publicly available data from ProteomicsDB. Because missing values in these datasets typically indicate non-detection rather than confirmed absence, we selected the 10 human cell lines with the greatest overlap in detected proteins with the HEK-293 prey/bait list. This strategy ensured sufficient coverage to enable meaningful comparison of expression levels across cell types, while minimizing the confounding effects of missing data. For each bait, expression levels were retrieved from normalized expression profiles provided by ProteomicsDB. Expression values were extracted for all 11 cell lines (HEK-293 plus 10 others), and a heatmap was generated to visualize the expression of all selected baits across cell lines. To quantify expression consistency, we excluded four baits—CALR3, CYP2C1_sigseq, HIST1H2BG, and SV40_NLS—from the statistical analysis. These baits were excluded because they are either not endogenously expressed in human cell lines (CALR3), correspond to non-human constructs (CYP2C1_sigseq and HIST1H2BG, which are derived from rabbit and mouse, respectively), or represent synthetic elements (SV40_NLS). Among the remaining 46 baits, we calculated the number of cell lines in which each bait was expressed (non-zero value). We then determined the percentage of baits expressed in all cell lines and the percentage expressed in at least half.

Simulation analysis and adjusted prey–bait matrices

To assess how bait selection methods perform in different cellular contexts, we simulated prey–bait interaction matrices for multiple cell lines using expression data from ProteomicsDB. Since protein abundance varies across cell lines, we adjusted the prey–bait interaction values based on expression ratios between HEK-293 and each target cell line. This adjustment accounted for differences in protein availability, allowing us to model how interactome structures might change across cellular environments.

For each cell line, we generated an adjusted prey–bait matrix by scaling interactions according to the relative expression levels of both the bait and the prey. If a bait or prey was missing in a given cell line, its interactions were excluded to ensure biological relevance. In more detail, the adjusted prey–bait interaction matrix for each cell line is computed using the relative expression ratios of baits and preys between HEK-293 and the target cell line. The formula for adjusting interaction values is:

$$A'_{\mathrm{ij}} = A_{\mathrm{ij}} \times \left(\frac{E_{\mathrm{target,\,i}}}{E_{\mathrm{HEK-293,\,i}}}\right) \times \left(\frac{E_{\mathrm{target,\,j}}}{E_{\mathrm{HEK-293,\,j}}}\right)$$

Where:

- A_{ij} is the original prey–bait interaction value in HEK-293.
- A'_{ij} is the adjusted interaction value for the target cell line.
- $E_{\mathrm{HEK\text{-}293,i}}$ and $E_{\mathrm{target,i}}$ are the expression levels of bait i in HEK-293 and the target cell line, respectively.

- $E_{\text{HEK-293,j}}$ and $E_{\text{target,j}}$ are the expression levels of prey j in HEK-293 and the target cell line, respectively.

To evaluate how well bait selection strategies generalize across cell lines, bait panels selected in HEK-293 by GENBAIT and 10 other methods were applied to the simulated prey–bait matrices of other cell lines, alongside a random baseline. Bait subset sizes ranged from 30 to 80, with 10 random seeds for each. The effectiveness of selection strategies was assessed using defined metrics (Supplementary Note 1).

Correlation between GNEBAIT performance and prey expression similarity

To test whether GENBAIT performance is influenced by overall similarity in expression profiles, we also calculated the Pearson correlation between each cell line's normalized prey expression profile and that of HEK-293. These correlations were then compared to GENBAIT's mean NMF Pearson scores per cell line, which were computed by averaging the NMF-based component similarity scores across bait subset sizes of 30–80, and across 10 random seeds. The resulting correlation plot allowed us to assess whether GENBAIT's effectiveness is associated with prey expression similarity between cell lines.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The SAINT files used in this study are available from the following sources: dataset 1 (Supplementary Table 2 of [https://static-content.springer.com/esm/art%3A10.1038%2Fs41586-021-03592-2/MediaObjects/41586_2021_3592_MOESM2_ESM.zip]), dataset 2 ([https://www.cell.com/cms/10.1016/j.molcel.2017.12.020/attachment/a4771708-1145-4272-86c0-aa2a0ba0e278/mmc2.xlsx]), and dataset 3 ([https://massive.ucsd.edu/ProteoSAFe/DownloadResultFile?file=f.MSV000090684%2Fother%2FDyakov_Table_3.xlsx&forceDownload=true]). Gene Ontology annotations were obtained from the GO Annotation File (GAF) format ([https://geneontology.org/docs/go-annotation-file-gaf-format-2.1/]). The source data underlying all main and Supplementary Figures are provided as a Source data file available at [https://doi.org/10.5281/zenodo.16580130][79].

Code availability

The GENBAIT Python package is available at Github [https://github.com/camlab-bioml/genbait] and permanently archived at Zenodo [https://doi.org/10.5281/zenodo.16579445][80] All code required to reproduce the analyses and figures in this study are available at https://github.com/camlab-bioml/genbait_reproducibility and permanently archived at Zenodo: https://doi.org/10.5281/zenodo.16580131 [81].

References

1. Christopher, J. A. et al. Subcellular proteomics. *Nat. Rev. Methods Prim.* **1**, 32 (2021).
2. Brunet, S. Organelle proteomics: looking at less to see more. *Trends Cell Biol.* **13**, 629–638 (2003).
3. Thul, P. J. et al. A subcellular map of the human proteome. *Science* **356**, eaal3321 (2017).
4. Go, C. D. et al. A proximity-dependent biotinylation map of a human cell. *Nature* **595**, 120–124 (2021).
5. Orre, L. M. et al. SubCellBarCode: proteome-wide mapping of protein localization and relocalization. *Mol. Cell* **73**, 166–182.e7 (2019).
6. Itzhak, D. N., Tyanova, S., Cox, J. & Borner, G. H. Global, quantitative and dynamic mapping of protein subcellular localization. *Elife* **5**, e16950 (2016).
7. Zhu, Y. et al. Cross-link assisted spatial proteomics to map sub-organelle proteomes and membrane protein topologies. *Nat. Commun.* **15**, 3290 (2024).
8. Thumuluri, V., Almagro Armenteros, J. J., Johansen, A. R., Nielsen, H. & Winther, O. DeepLoc 2.0: multi-label subcellular localization prediction using protein language models. *Nucleic Acids Res.* **50**, W228–W234 (2022).
9. Christoforou, A. et al. A draft map of the mouse pluripotent stem cell spatial proteome. *Nat. Commun.* **7**, 9992 (2016).
10. Desjardins, M. ER-mediated phagocytosis: a new membrane for new functions. *Nat. Rev. Immunol.* **3**, 280–291 (2003).
11. Borner, G. H. H. Organellar maps through proteomic profiling—a conceptual guide. *Mol. Cell. Proteom.* **19**, 1076–1087 (2020).
12. Samavarchi-Tehrani, P., Samson, R. & Gingras, A.-C. Proximity dependent biotinylation: key enzymes and adaptation to proteomics approaches. *Mol. Cell. Proteom.* **19**, 757–773 (2020).
13. Samavarchi-Tehrani, P., Abdouni, H., Samson, R. & Gingras, A.-C. A versatile lentiviral delivery toolkit for proximity-dependent biotinylation in diverse cell types. *Mol. Cell. Proteom.* **17**, 2256–2269 (2018).
14. Roux, K. J., Kim, D. I., Raida, M. & Burke, B. A promiscuous biotin ligase fusion protein identifies proximal and interacting proteins in mammalian cells. *J. Cell Biol.* **196**, 801–810 (2012).
15. Kim, D. I. et al. Probing nuclear pore complex architecture with proximity-dependent biotinylation. *Proc. Natl. Acad. Sci. USA* **111**, E2453-61 (2014).
16. Choi, H. et al. SAINT: probabilistic scoring of affinity purification–mass spectrometry data. *Nat. Methods* **8**, 70–73 (2011).
17. Choi, H. et al. Analyzing protein-protein interactions from affinity purification-mass spectrometry data with SAINT. *Curr. Protoc. Bioinform.* **39**, 8.15 (2012).
18. Teo, G. et al. SAINTexpress: improvements and additional features in significance analysis of INTeractome software. *J. Proteom.* **100**, 37–43 (2014).
19. Trinkle-Mulcahy, L. Recent advances in proximity-based labeling methods for interactome mapping. *F1000Res* **8**, 135 (2019).
20. Branon, T. C. et al. Efficient proximity labeling in living cells and organisms with TurboID. *Nat. Biotechnol.* **36**, 880–887 (2018).
21. Roux, K. J., Kim, D. I., Burke, B. & May, D. G. BioID: a screen for protein-protein interactions. *Curr. Protoc. Protein Sci.* **91**, 19.23.1–19.23.15 (2018).
22. Dhillon, I. S. & Sra, S. Generalized nonnegative matrix approximations with Bregman divergences. In *Proc. 19th International Conference on Neural Information Processing Systems* 283–290 (MIT Press, 2005).
23. Youn, J.-Y. et al. High-density proximity mapping reveals the subcellular organization of mRNA-associated granules and bodies. *Mol. Cell* **69**, 517–532.e11 (2018).
24. Raudvere, U. et al. g:Profiler: a web server for functional enrichment analysis and conversions of gene lists (2019 update). *Nucleic Acids Res.* **47**, W191–W198 (2019).
25. Dyakov, B. J. A. et al. Spatial proteomic mapping of human nuclear bodies reveals new functional insights into RNA regulation. Preprint at *bioRxiv* 2024.07.03.601239. https://doi.org/10.1101/2024.07.03.601239 (2024).
26. Antonicka, H. et al. A high-density human mitochondrial proximity interaction network. *Cell Metab.* **32**, 479–497.e9 (2020).
27. Gupta, G. D. et al. A dynamic protein interaction landscape of the human centrosome-cilium interface. *Cell* **163**, 1484–1499 (2015).
28. Guyon, I. M. & Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **3**, 1157–1182 (2003).
29. Yang, P., Huang, H. & Liu, C. Feature selection revisited in the single-cell era. *Genome Biol.* **22**, 321 (2021).
30. Lualdi, M. & Fasano, M. Statistical analysis of proteomics data: a review on feature selection. *J. Proteom.* **198**, 18–26 (2019).

31. Banzhaf, W. *Genetic Programming: an Introduction* (Elsevier Science, 1998).
32. Mitchell, M. *An Introduction to Genetic Algorithms* (MIT Press, 1998).
33. Manning, T., Sleator, R. D. & Walsh, P. Naturally selecting solutions. *Bioengineered* **4**, 266–278 (2013).
34. Miralavy, I., Bricco, A. R., Gilad, A. A. & Banzhaf, W. Using genetic programming to predict and optimize protein function. *PeerJ Phys. Chem*. **4**, e24 (2022).
35. HajiHosseinKhani, S., Lashkari, A. H. & Mizani Oskui, A. Unveiling vulnerable smart contracts: toward profiling vulnerable smart contracts using genetic algorithm and generating benchmark dataset. *Blockchain Res. Appl*. **5**, 100171 (2024).
36. Kuhn, H. W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q*. **2**, 83–97 (1955).
37. Oliveto, P. S., Paixão, T., Pérez Heredia, J., Sudholt, D. & Trubenová, B. How to escape local optima in black box optimisation: when non-elitism outperforms elitism. *Algorithmica* **80**, 1604–1633 (2018).
38. Fisher, R. A. *Statistical Methods for Research Workers*, 11th Edn Rev (Oliver and Boyd, 1925).
39. Ma, S. & Huang, J. Penalized feature selection and classification in bioinformatics. *Brief. Bioinform*. **9**, 392–403 (2008).
40. Jovic, A., Brkic, K. & Bogunovic, N. A review of feature selection methods with applications. In *Proc. 38th International Convention on Information and Communication Technology, Electronics and Microelectronics*(*MIPRO*) 1200–1205 (IEEE, 2015).
41. Friedman, J. H. Greedy function approximation: a gradient boosting machine. (1999).
42. Natekin, A. & Knoll, A. Gradient boosting machines, a tutorial. *Front Neurorobot*. **7**, 21 (2013).
43. Chen, T. & Guestrin, C. XGBoost. In *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 785–794 (ACM, 2016).
44. *The Shapley Value* (Cambridge University Press, 1988).
45. Kraev, E., Koseoglu, B., Traverso, L. & Topiwalla, M. Shap-select: lightweight feature selection using SHAP values and regression. Preprint at https://doi.org/10.48550/arXiv.2410.06815 (2024).
46. Pearson, K. & Galton, F. V. I. I. Note on regression and inheritance in the case of two parents. *Proc. R. Soc. Lond*. **58**, 240–242 (1895).
47. Singhal, A. Modern information retrieval: a brief overview. *IEEE Data Eng. Bull*. **24**, 35–43 (2001).
48. Kullback, S. & Leibler, R. A. On information and sufficiency. *Ann. Math. Stat*. **22**, 79–86 (1951).
49. Hubert, L. & Arabie, P. Comparing partitions. *J. Classif*. **2**, 193–218 (1985).
50. Manning, C. D., Raghavan, P. & Schütze, H. *Introduction to Information Retrieval* (Cambridge University Press, 2008).
51. Murphy, A. H. The finley affair: a signal event in the history of forecast verification. *Weather Forecast* **11**, 3–20 (1996).
52. Cover, T. & Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **13**, 21–27 (1967).
53. Traag, V. A., Waltman, L. & van Eck, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep*. **9**, 5233 (2019).
54. Yu, G., Sapiro, G. & Mallat, S. Solving inverse problems with piecewise linear estimators: from Gaussian mixture models to structured sparsity. *IEEE Trans. Image Process*. **21**, 2481–2499 (2012).
55. Lautenbacher, L. et al. ProteomicsDB: toward a FAIR open-source resource for life-science research. *Nucleic Acids Res*. **50**, D1541–D1552 (2022).
56. Lin, C.-J. Projected gradient methods for nonnegative matrix factorization. *Neural Comput*. **19**, 2756–2779 (2007).
57. Zhai, Y., Song, W., Liu, X., Liu, L. & Zhao, X. A chi-square statistics based feature selection method in text classification. In *Proc. IEEE 9th International Conference on Software Engineering and Service Science*(*ICSESS*) 160–163 (IEEE, 2018).
58. Pedregosa, F. et al. Scikit-learn: machine learning in Python. *J. Mach. Learn. Res*. **12**, 2825–2830 (2011).
59. Elssied, N. O. F., Ibrahim, O. & Osman, A. H. A novel feature selection based on one-way anova f-test for e-mail spam classification. *Res. J. Appl. Sci. Eng. Technol*. **7**, 625–638 (2014).
60. Vergara, J. R. & Estévez, P. A. A review of feature selection methods based on mutual information. *Neural Comput. Appl*. **24**, 175–186 (2014).
61. Muthukrishnan, R. & Rohini, R. LASSO: a feature selection technique in predictive modeling for machine learning. In *Proc. IEEE International Conference on Advances in Computer Applications*(*ICACA*) 18–20. https://doi.org/10.1109/ICACA.2016.7887916 (IEEE, 2016).
62. Defazio, A., Bach, F. & Lacoste-Julien, S. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. *Adv. Neural Inf. Process. Syst*. **2**, 1646–1654 (2014).
63. Zou, H. & Hastie, T. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B Stat. Methodol*. **67**, 301–320 (2005).
64. Kursa, M. B. & Rudnicki, W. R. The all relevant feature selection using random forest. Preprint at https://doi.org/10.48550/arXiv.1106.5112 (2011).
65. Xu, Z., Huang, G., Weinberger, K. Q. & Zheng, A. X. Gradient boosted feature selection. In *Proc. 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 522–531 (Association for Computing Machinery, 2014).
66. Chen, C. et al. Improving protein-protein interactions prediction accuracy using XGBoost feature selection and stacked ensemble classifier. *Comput. Biol. Med*. **123**, 103899 (2020).
67. Falcon, W. & team, T. P. L. PyTorch Lightning. Preprint at. https://doi.org/10.5281/zenodo.13254264 (2024).
68. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. Preprint at https://arxiv.org/abs/1412.6980 (2017).
69. Koza, J. & Poli, R. Genetic Programming. in *Search Methodologies* 127–164. https://doi.org/10.1007/0-387-28356-0_5. (2005)
70. Fortin, F.-A. De Rainville, F.-M. Gardner, M.-A. Parizeau, M. Gagné, C. DEAP: evolutionary algorithms made easy. *J. Mach. Learn. Res*. **13**, 2171–2175 (2012).
71. Harris, C. R. et al. Array programming with NumPy. *Nature* **585**, 357–362 (2020).
72. McKinney, W. Data structures for statistical computing in Python. 56–61. https://doi.org/10.25080/Majora-92bf1922-00a (2010).
73. Virtanen, P. et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272 (2020).
74. Traag, V. et al. vtraag/leidenalg: 0.10.0. Preprint at https://doi.org/10.5281/zenodo.8147844 (2023).
75. Csardi, G. & Nepusz, T. The igraph software package for complex network research. *InterJournal, Complex Systems* **1695**, 1–9 (2006).
76. Ashburner, M. et al. Gene Ontology: tool for the unification of biology. *Nat. Genet* **25**, 25–29 (2000).
77. Aleksander, S. A. et al. The Gene Ontology Knowledgebase in 2023. *Genetics* **224**, iyad031 (2023).
78. Hagberg, A. A., Schult, D. A., Swart, P. & Hagberg, J. M. Exploring Network Structure, Dynamics, and Function using NetworkX. *Proceedings of the Python in Science Conference* (2008).
79. Kasmaeifar, V. camlab-bioml/genbait_reproducibility: GENBAIT Reproducibility – 'Computational design and evaluation of optimal bait sets for scalable proximity proteomics'. https://doi.org/10.5281/zenodo.16762583 (2025).
80. Kasmaeifar, V. camlab-bioml/genbait: GENBAIT v1.0 – Code for "Computational design and evaluation of optimal bait sets for scalable proximity proteomics". https://doi.org/10.5281/zenodo.16579445 (2025).
81. Kasmaeifar, V. camlab-bioml/genbait_reproducibility: GENBAIT Reproducibility – 'Computational design and evaluation of optimal

bait sets for scalable proximity proteomics'. https://doi.org/10.5281/zenodo.16580131 (2025).

Acknowledgements

We thank High-Fidelity Science Communications for manuscript editing. We thank members of the Gingras and Campbell labs for helpful discussions. This work was supported by a Canadian Institutes of Health Research (CIHR) Project Grant (PJT-185987), a National Science and Engineering Research Council of Canada (NSERC) Discovery grant (RGPIN-2019-06297) to A.C.G. and a Canadian Foundation for Innovation JELF award, a Canadian Institutes of Health Research (CIHR) Project Grant (PJT-175270) and a National Science and Engineering Research Council of Canada (NSERC) Discovery grant (RGPIN-2020-04083) to K.R.C. A.C.G. is supported by a Canada Research Chair (Tier 1) in Functional Proteomics and the Lou Siminovitch Mount Sinai 100 Chair. K.R.C. is supported by a Canada Research Chair (Tier 2) in Machine Learning for Translational Biomedicine. S.S. was supported by the University of Toronto Open Fellowships.

Author contributions

Project conception: A.-C.G., V.K., and K.R.C. Result interpretation and manuscript writing: V.K., K.R.C., and A.-C.G. Data analysis and software development: V.K. and K.R.C. Visualization: V.K. and S.S.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-025-64383-1.

Correspondence and requests for materials should be addressed to Anne-Claude Gingras or Kieran R. Campbell.

Peer review information *Nature Communications* thanks the anonymous reviewers for their contribution to the peer review of this work. A peer review file is available.