ICAT: A Novel Algorithm to Robustly Identify Cell States Following Perturbations in Single Cell Transcriptomes

Dakota Y. Hawkins[1,2], Daniel T. Zuch[2,3], James Huth[2], Nahomie Rodriguez-Sastre[2], Kelley R. McCutcheon[2], Abigail Glick[2], Alexandra T. Lion[2,3], W. Evan Johnson[1,6], and Cynthia A. Bradham[1,2,3,5*]

[1]Bioinformatics Program, [2]Biology Department, [3]Program in Molecular and Cellular Biology and Biochemistry, and [5]Biological Design Center, Boston University, Boston, MA, USA

[6]Division of Computational Biomedicine, Boston University School of Medicine, Boston, MA, USA

*Author for correspondence; e-mail: cbradham@bu.edu

Author e-mails: DYH, dyh0110@bu.edu; DTZ, dtzuch@bu.edu; JH, jhuth@bu.edu; NRS, chantal1@bu.edu; KRM, kmccutch@bu.edu; AG, aglick18@bu.edu; WEJ, wej@bu.edu

KEY WORDS: single cell, scRNA-seq, clustering, cell states, perturbation, cell type, semi-supervised, self-supervised

Abstract

The detection of distinct cellular identities is central to the analysis of single-cell RNA sequencing experiments. However, analyzing datasets with experimental treatments using common methods often mis-identifies cell states. Here we present the novel algorithm ICAT that employs self-supervised feature weighting and semi-supervised clustering to accurately resolve cell states across heterogeneous conditions. Using various datasets, we show ICAT better resolves cell states compared to integration workflows, while remaining robust to signal and perturbation severity. We empirically validate ICAT and show only ICAT identifies a perturbation-unique cell state, demonstrating that ICAT offers a necessary improvement in evaluating cellular response to perturbation.

Background

From deconstructing tumor cell compositions [1], to reconstructing developmental pathways [2], single-cell RNA sequencing (scRNA-seq) has revolutionized scientists' ability to explore complex tissues and systems. Clustering of individual cells is central to the analysis of scRNA-seq data, in which cell states are identified by grouping cells with similar gene expression profiles [3]. Whether to ascertain the effect of drug treatment on cancer viability, or to infer the mechanistic role of a given gene in tissue specification, perturbation experiments have long been used to dissect and understand complex biological systems. Oftentimes this analysis is formalized by identifying changes in the abundance of cell identities between treatments [3–5]. Inherently, such analyses are based on the assumption that cell states are readily identified and matched between treatments. However, matching identities across perturbing treatments presents particular issues with scRNA-seq datasets due to dominating treatment effects, especially among cell states that are already similar [4]. Thus, it is typical that scRNA-seq datasets segregate by treatments rather than cell identities (Fig. S1) [3,4,6,7]. This challenge presents significant difficulties in identifying and tracking specific cell identities across treatments.

Common techniques for identifying cell states in scRNA-seq perturbation experiments rely on either reference datasets to map cells onto known cell identities [8,9], or an integration algorithm to minimize differences between control and perturbed cells by transforming cells into a shared space prior to clustering cells [3,10]. The first scenario is problematic for non-model systems, where such atlases may not exist, and is inappropriate when the goal of a given experiment is discovery of novel cell identities or subtypes; for these reasons, the latter approach is often preferred. Toward that end, scRNA-seq algorithms have been developed that

61 broadly integrate heterogeneous datasets [3,11,12]. Briefly, integration algorithms identify a
62 shared space between separate datasets, then subsequently project cells into this embedding.
63 While promising for combining datasets across modalities, integration methods can smooth over
64 and eliminate biological signals, leading to incorrect identification and classification of cell states
65 [3,13,14].

66 We therefore developed a two-step algorithm to robustly and sensitively **I**dentify **C**ell
67 states **A**cross **T**reatments (ICAT) in scRNA-seq data. By employing self-supervised feature
68 weighting, followed by semi-supervised clustering, ICAT accurately identifies cell states across
69 scRNA-seq perturbation experiments with high accuracy. ICAT does not require prior
70 knowledge of marker genes or extant cell states, is robust to perturbation severity, and identifies
71 cell states with higher accuracy than leading integration workflows within both simulated and
72 real scRNA-seq perturbation experiments, while operating at lower signal resolutions.

73 Results

74 **Discriminating cell states across perturbations in single-cell RNAseq data.**

75 To overcome the inability of current clustering algorithms to accurately detect cell
76 identities in perturbation experiments (Fig. S1), we developed a two-step algorithm (Fig. 1A).
77 Using the Louvain method [15] by default, the first step separately clusters control cells to
78 produce initial cluster labels for each cell. Cluster labels are then used to weight genes by their
79 ability to predict the previously generated labels using Neighborhood Component Feature
80 Selection (NCFS) [16]. NCFS weights genes by using gradient ascent to maximize a regularized
81 leave-one-out classification accuracy metric. By the nature of regularization, most gene weights
82 converge to zero, leaving only the most predictive genes with weights greater than 1. Applying
83 the learned gene weights to the original expression matrix with both control and treated cells
84 transforms the matrix into a “cluster-defined” space, where the distances between cells are

85 dominated by highly weighted marker genes. After this transformation, ICAT retains previously
86 identified cluster labels for control cells, then performs semi-supervised Louvain community
87 detection [15,17], whereby control labels are held immutable (Fig. 1A).
88 This strategy achieves two goals: 1. ICAT makes comparing treatment effects on cell
89 states straightforward. By implementing semi-supervised clustering with immutable control
90 labels, ICAT ensures identities discovered in control cells are retained when treated cells are
91 introduced. 2. ICAT produces easily interpretable embeddings. Because NCFS is a sparse
92 method, it overtly separates informative and non-informative genes. Thus, the expression
93 patterns of a few highly weighted genes directly account for cell location in the NCFS-
94 transformed space. This is a noted benefit over typical integration methods, where
95 transformations are obscured behind neighbor-based approaches [11,12]. These two features
96 together produce interpretable results that ease compositional analysis between treatments.
97 While similar in approach, ICAT provides several advantages compared to identifying
98 shared cell states via marker gene mapping: first, ICAT does not require previously known
99 marker genes, making ICAT appropriate for situations with previously uncharacterized cell
100 states. Second, by clustering control and treated cells together, ICAT takes advantage of the
101 structure in both datasets compared to clustering both independently. This alleviates issues with
102 imperfect matches between marker genes. Third, by foregoing common marker gene presence-
103 absence comparisons to map cell types, ICAT accurately identifies activated states, which is
104 otherwise difficult. Finally, by generating cell state labels through a clustering step rather than a
105 classification approach, ICAT is able to group cells with treatment-specific phenotypes and
106 identify out-of-sample states.
107 One caveat of our approach is that if asymmetrical populations exist in treated cells, but
108 not controls, their identifying genes will likely be collapsed by the control-only feature-weighting
109 step. To avoid this artifact, ICAT can be optionally enabled to perform feature-weighting in all
110 treatment types to identify asymmetrical populations across conditions (Fig. 1B, Fig. 2A). To

111 reduce gene weights to a single gene-weight vector, only the maximum gene weight produced
112 across the conditions is retained. During semi-supervised clustering, labels for control cells are
113 retained during label initialization, while treated cells are re-initialized as singleton clusters.
114
115 **Evaluating performance across conditions using simulated datasets.**
116 To assess ICAT, we initially tested performance in simulated single-cell RNAseq data
117 (Fig. 2A-E). We simulated data using a zero-inflated Negative Binomial distribution to model
118 gene counts per Büttner and colleagues [14]. To simulate perturbations, we sampled a scalar
119 multiplier from a $\Gamma(2,2)$ distribution, then applied this multiplier to randomly selected genes as
120 perturbation targets. To generate distinct cell types, we randomly selected a separate set of
121 genes to act as markers for each population. Selected marker genes had their average
122 expression shifted by a scalar multiplier following a $\Gamma(3,3)$ distribution to create population
123 substructure. To test ICAT's ability to detect both asymmetrical populations and activated
124 states, we simulated five populations of cells: three that are present in both control and treated
125 cells, with one of these exhibiting elevated marker gene expression upon treatment and referred
126 to as "stimulated" (population C1 and P(C1)+), and two treatment-only populations (populations
127 P4 and P5) (Fig. 2A-B).
128 Using simulated data, we first assessed ICAT's robustness against increasing
129 perturbation severity, testing five conditions with varying proportions of perturbed genes,
130 ranging from 1% to 25% (Fig. 2B, Table S1). Next, we assessed performance with changing
131 resolutions between cell types by simulating 10 different experimental conditions with a varying
132 number of average marker genes per cell type. Mean marker gene numbers ranged from 10-
133 105 per population (0.67-7.0% of total genes) (Fig. 2C, Table S2). In both cases, each
134 experiment was simulated 15 times, for a total of 90 independent datasets in the first case (Fig.
135 2B) and 150 in the second case (Fig. 2C.). Finally, to test performance as cell type proportions
136 deviate from equality, we simulated three different experimental conditions with varying

137 distributions of each cell type. Experiments ranged from all cell types having 100 cells per
138 treatment, to the most disparate where the minimum and maximum population size per
139 treatment were 50 and 150, respectively (Fig. 2D, Table S3). To isolate the effect of differential
140 abundance, proportion simulations did not include an activated cell type, and instead were
141 composed of three shared populations and two treatment-unique populations (Fig. 2D). Each
142 experimental proportion was simulated 15 times.

143 When assessing ICAT's performance in simulated data, we considered three criteria
144 reflecting "good" clustering between control and treated cells: 1. Global label conservation: the
145 extent of mapping between cluster labels and known cell identities, which is ideally one-to-one
146 to reflect accurate detection of cell identities, was assessed with the Adjusted Rand Index [18]
147 (ARI). 2. Treatment mixing: the local homogeneity of shared cell types without undo isolation by
148 treatment status was assessed with the Local Inverse Simpson's Index [19] (LISI); 3. Unique
149 label separation: the accurate detection of populations-of-interest was measured using
150 population-specific F1 measures [13]. Such metrics were used to measure the ability to
151 accurately detect asymmetric populations that exist only in single conditions (F1-Unique), the
152 ability to discriminate stimulated cell states from non-stimulated analogues (F1-Stim), and the
153 identification of rare cell types (F1-Rare). ARI, LISI, and all F1 metrics were standardized
154 between 0 and 1, where 1 indicated stronger performance.

155

156 **Comparing performance to current state-of-the-art methods.**

157 We compared ICAT against two state-of-the-art integration methods: Seurat [11,20] and
158 Scanorama [12]. Seurat was chosen for its ubiquity in single-cell analysis, while Scanorama was
159 chosen due to its strong performance in benchmarks comparisons [13]. In both cases, datasets
160 were integrated between treatments, then the results were clustered via Louvain community
161 detection. To establish baseline performance, we also tested the "No Integration" scenario in
162 which cells were clustered using Louvain community detection without accounting for treatment

status. During clustering, we standardized nearest neighbor and resolution parameters for each method.

ICAT is robust to perturbation severity and better resolves cell states at lower resolutions than leading integration workflows.

When testing perturbation severity or variable marker gene numbers (Fig. 2F), ICAT outperformed both Scanorama and Seurat across all metrics while exhibiting highly stable performance across both increasing perturbation intensities and at varying cell-state resolutions (Fig. 2F). ICAT provides near-perfect treatment mixing, with LISI scores nearing 1 for all signal levels. In contrast, Scanorama and Seurat only approach 0.5 LISI scores at the highest signal levels, indicating that ICAT exhibits a substantially improved ability to correctly and robustly mix cell identities across treatments (Fig. 2F).

Scanorama offered comparable performance to ICAT in global label conservation (ARI) and the ability to detect asymmetrical populations (F1-Unique) (Fig. 2F), while ICAT was uniformly better at accurately distinguishing the stimulated population from its non-stimulated analogue (Fig. 2F, F1-Stim). Seurat underperformed both Scanorama and ICAT in label conservation and asymmetrical population detection (ARI, F1-Unique), and offered comparable treatment-mixing performance to Scanorama (LISI), indicating that while Seurat successfully mixes cells at lower perturbation severities, it does not accurately separate known populations relative to Scanorama or ICAT (Fig. 2F).

For each of the perturbation severity tests and with low marker gene numbers, Seurat was surprisingly outperformed by the no-integration scenario for ARI (Fig. 2F) which can be explained by Seurat's general under-clustering of cells (Fig. S2A). Scanorama exhibited a similar drop in ARI performance at low marker numbers, but consistently outperformed no integration (Fig. 2F). This shows that Seurat and Scanorama are more reliable with higher

188 numbers of discriminating marker genes, whereas ICAT is robust to low signal strength and the
189 most reliable approach overall.
190
191 **ICAT accurately detects rarer populations.**
192 ICAT was unaffected by differences in cell identity proportions as assessed by global
193 and rare population label conservation via ARI and the F1-Rare metric, respectively, while
194 Seurat and Scanorama each exhibited decreased performance as the disparity in proportions
195 increased (Fig. 2F, left). The performance of ICAT was robust and stronger than both
196 Scanorama and Seurat in each measure, while Scanorama outperformed Seurat (Fig. 2G, left).
197
198 **ICAT improves and stabilizes performance of integration workflows.**
199 Outside of requiring a data matrix, ICAT is agnostic to input data. Accordingly, we also
200 evaluated clustering performance for two extended methods: $ICAT_{Seurat}$ and $ICAT_{Scan}$, whereby
201 cells were first integrated using Seurat or Scanorama, respectively, then clustered with ICAT. In
202 this way, ICAT can be utilized either as an alternative to integration methods, or as an additional
203 step after integration.
204 Compared to Seurat, $ICAT_{Seurat}$ led to significant improvements in all three simulated
205 experiments, leading to greater robustness to perturbation (Fig. 2H1, left), increased signal
206 sensitivity (Fig. 2H2, left), and better isolation of rare cells (Fig. 2G, middle). When compared to
207 Scanorama, $ICAT_{Scan}$ better isolated rare cells (Fig. 2G, right), and improved performance in
208 some metrics in perturbation and signal experiments. Gains were more modest compared to
209 $ICAT_{Seurat}$, primarily since Scanorama alone already exhibits strong performance (Fig. 2H1-2,
210 right). In particular, $ICAT_{Scan}$ substantially improved treatment-mixing (LISI) across conditions
211 compared to Scanorama alone (Fig. 2H, right). Interestingly, at both signal extremes, stand-
212 alone ICAT scored higher in the F1-Stim metric than either combined or original integration

213 methods (Fig. 2H2), indicating that ICAT better preserves discriminatory signals compared to
214 either integration algorithms alone or extended.
215 In summary, the results from simulated data demonstrate that ICAT offers robust and
216 accurate cluster identification performance that exceeds two current state-of-the-art integration
217 algorithms across a range of simulated conditions including severe perturbations, low marker
218 numbers, and variations in cell abundance. Further analysis shows that combining ICAT with
219 two common integration algorithms generally improves their performance.
220
221 **Measuring performance in real datasets.**
222 To assess ICAT's ability to correctly cluster cells in real perturbation experiments, we
223 made use of three publicly available datasets (Table S4). While no true benchmark datasets
224 currently exist for defining cell identities in perturbation experiments, we selected datasets that
225 either contain genotyped cell lines, or are from well-studied cell types with confidently labeled
226 cells. The first dataset, by Tian et al., was created as a standard to evaluate methods developed
227 for scRNA-seq data analysis [21]. To evaluate ICAT, we used their CellMix dataset that was
228 produced by creating various mixtures of nine cells from combinations of H2228, HCC827, and
229 H1975 adenocarcinoma cell lines, sequencing the cell mixtures, then downsampling the
230 resulting counts to produce pseudo-cells that are in-line with true single-cell data. While the Tian
231 dataset was developed to test integration across batch effects, we reasoned that variance
232 introduced by batch effects is similar to perturbation-induced variation. To better mimic distinct
233 phenotypes, we selected only four cell "types", consisting of pure H2228, HCC827, and H1975
234 cells, along with a single mixture with equal proportions of each cell line, denoted "Mixed" (Fig.
235 S3, Table S4). The second dataset from Kagohara et al. [7] offers a simple perturbation
236 experiment in which three genotyped cell lines of head and neck squamous carcinoma cells
237 were treated with either PBS (control) or the chemotherapeutic cetuximab (Table S4). Finally,
238 we assessed performance in a peripheral blood mononuclear cell dataset collected by Kang et

239 al.[4] with eight cell types in control and IFN-β-stimulated conditions (Table S4). Prior to analysis,
240 each dataset was preprocessed by normalizing cell counts and selecting highly variable genes
241 using *Scanpy*. To ensure each tested algorithm analyzed the same input data, we provided the
242 same *Scanpy* preprocessed data to each method.

243 Because NCFS feature-weighting is a computationally intensive task, and since the
244 Kagohara and Kang datasets each contain > 20,000 cells, it was necessary to subselect cells
245 for feature weighting. To this end, ICAT selects a representative sample of cells via submodular
246 optimization using the *apricot* [22] Python package. Submodular optimization via *apricot* was
247 able to effectively represent the data space using only a small number of cells (Fig. S4).

248 To measure ICAT's performance in these datasets, we again evaluated clustering quality
249 using ARI and LISI metrics to measure global label conservation and treatment status mixing,
250 respectively. Because cell identity is resolution dependent and likely imperfect in real datasets
251 [3,6], we also calculated the label-free Davies-Bouldin (DB) metric to assess the density of
252 clusters produced by each method [23]. The DB metric measures similarity between a cluster
253 and the next most similar cluster. To facilitate comparison, each score was scaled between 0
254 and 1 such that 1 represents better clustering while 0 represents worse clustering.

255

256 **ICAT produces higher quality clusters in real datasets compared to integration methods.**

257 In the Tian dataset, ICAT outperformed both existing integration methods in LISI and DB
258 metrics and was comparable to both in ARI, in which ICAT was slightly worse than Seurat and
259 Scanorama (Fig. 3A-B, Table S5). ICAT outperformed both integration methods in the Kagohara
260 and Kang datasets (Fig. 3A-B). Aside from ARI in the Kagohara dataset, in which Scanorama
261 and ICAT scored equally well (Fig. 3B), ICAT produced the highest scores for ARI, LISI, and DB
262 metrics (Fig. 3B). Overall, ICAT generated higher quality clusters in real datasets compared to
263 existing integration algorithms (Fig. 3A-B, Table S5).

264

265 **Combining integration algorithms with ICAT produced higher quality clusters in real**

266 **datasets.**

267 Compared to Scanorama, $ICAT_{Scan}$ led to greater ARI, LISI and DB scores across the

268 three analyzed datasets (Fig. 3C, Table S5), aside from ARI in the Kagohara dataset where

269 Scanorama and $ICAT_{Scan}$ scored equally well (Fig. 3C). For Seurat, $ICAT_{Seurat}$ resulted in

270 improvements that exhibit dataset and metric dependence: cell states were more readily

271 identified in the Kagohara dataset, shown by the dramatic increase in ARI scores, and clusters

272 were tighter and better separated as shown by large increases in DB scores (Fig. 3C). In

273 contrast, ARI scores were maintained or marginally reduced with $ICAT_{Seurat}$ compared to

274 standalone Seurat for the Tian and Kang datasets. Overall, these results show that ICAT not

275 only offers strong stand-alone performance, but also generally improves existing integration

276 workflows to better cluster cells in scRNA-seq data.

277

278 **Validating ICAT predictions in perturbed sea urchin embryos.**

279 To empirically test ICAT predictions on cell state compositions following perturbation, we

280 generated scRNA-seq data using the SMART-Seq2 protocol [24]. Specifically, we isolated and

281 sequenced the skeletogenic lineage known as primary mesenchyme cells (PMCs) at 18 hours

282 post-fertilization, comparing cells from embryos subjected to chemical treatments that perturb

283 skeletal patterning, specifically, MK-886 (MK) and chlorate [25–27]. PMCs are of interest

284 because this population of cells receives patterning cues during development that cause their

285 diversification into subsets; since MK and chlorate each perturb distinct patterning cues, we

286 reasoned that these differences would be reflected in PMC subpopulation composition [27–30].

287 After quality control and cell filtering, we retained expression data for 435 cells (147 control, 134

288 MK886, and 154 chlorate treated cells).

289

290 **ICAT accurately predicts subpopulation response to external perturbation *in vivo.***

291 ICAT clustered the combined PMC expression profiles into five distinct cell states (Fig.
292 4A1). The impacts of MK exposure were more dramatic than those of (Fig. 4A2); we therefore
293 focused on the effects of MK. Two clusters showed stark compositional differences between
294 control and MK-treated cells, in that cluster 3 lacks MK cells compared to controls, while
295 clusters 2 and 4 show significant enrichment for MK cells (G-test fdr < 0.05, post hoc Fisher's
296 Exact test fdr < 0.05; Fig. 4A2, Table S6-8). Since MK-treated cells dominated cluster 4 such
297 that it appears to be an asymmetric population, we more closely examined clusters 3 and 4.
298 Investigating highly predictive genes identified by ICAT found that cluster 3 and 4 were
299 distinguished by inverse expression of genes *pks2* and *sm50* (Fig. 4A3-4), such that cluster 3
300 displayed a *sm50+/pks2-* phenotype, while cluster 4 exhibited a reciprocal *sm50-/pks2+* pattern.

301 To experimentally validate ICAT's predicted enrichment of the *sm50-/pks2+* cell state
302 and depletion of *sm50+/pks2* cells, in MK886-treated PMCs, we performed single molecule
303 fluorescence *in situ* hybridization (FISH) [31] to quantify expression levels for both genes in
304 DMSO (control) and MK-treated embryos [31] (Fig. 4B, Fig. S5). To accurately map FISH
305 signals to PMCs, embryos were stained for PMCs using the PMC-specific 6a9 antibody. Using
306 Ilastik [32],we trained a random forest classifier (out-of-bag error = 0.14) to predict and segment
307 individual PMCs in each confocal image stack (Fig. 4B1-2, Fig. S6). Following methods for
308 relative FISH quantification from the original paper [31], we calculated cell-level expression
309 values for each PMC by first restricting fluorescence signal to PMC-labeled voxels, then
310 calculated the average relative intensity for each gene in every PMC (Fig. 4B3-4, Fig. S5).
311 Across embryos, we labeled individual PMCs as *sm50+* or *sm50-* if their standardized *sm50*
312 expression values were above the 75th or below the 25th percentiles of all PMCs, respectively
313 (Fig. S7). The procedure was repeated for *pks2* to determine *pks2+* and *pks2-* PMCs. We
314 found that MK-treated embryos displayed both a significant loss of the *sm50+/pks2-* phenotype,
315 as well as a significant enrichment of the *sm50-/pks2+* cell state relative to controls (adjusted p

316 < 0.05; Fig. 4B5-6, Table S6-9): verifying the ability of ICAT to correctly identify affected cell
317 states in scRNA-seq datasets from perturbed samples.
318
319 **Integration methods fail to correctly identify affected cell states.**
320 When the PMC scRNA-seq dataset were integrated using either Seurat or Scanorama,
321 each method failed to identify the enriched *sm50-/pks2+* cell state in MK-treated embryos, and
322 instead incorrectly merged those cells with other cell states (Fig. 4C, Fig. S8). In fact, Seurat
323 produced incorrect conclusions by failing to identify any dependence of cell state composition
324 on treatment status (G-test fdr > 0.05; Fig. S8A2, Table S10). Scanorama, on the other hand,
325 correctly identified the depletion of *sm50+/pks2-* PMCs in MK embryos (Fig S8B2-3, Table S11-
326 12; G-test fdr < 0.05, post hoc Fisher's Exact test fdr < 0.05), but importantly, failed to identify
327 *sm50-/pks2+* enrichment in MK embryos (Fig S8B2-3, Table S11-12; G-test fdr > 0.05). ICAT is
328 the only method among these three to correctly identify each cell state that is perturbed by MK
329 exposure. These results demonstrate that ICAT significantly improves the computational
330 capability to accurately detect and describe cell state-level responses to perturbation in scRNA-
331 seq data.

332 Discussion

333 When comparing the performance of ICAT to state-of-the-art integration algorithms in
334 simulated datasets, we found that ICAT performs more robustly in response to perturbation
335 severity, readily identifies cell states at lower signal levels, better isolates asymmetrical and
336 activated cell states, and more readily identifies rare cell populations when compared to either
337 Seurat or Scanorama integration workflows. ICAT also outperforms Seurat and Scanorama in
338 three benchmark datasets, further solidifying ICAT's ability to better identify cell states across
339 biological conditions compared to either integration method alone. By testing ICAT on datasets

340 with > 20,000 cells [4,7], we demonstrated ICAT's ability to analyze the large datasets,
341 necessary for any practical scRNA-seq tool. Depending on the experimental system,
342 researchers may still elect to use integration algorithms. Excitingly, the choice between ICAT
343 and integration methods is not mutually exclusive, and we show that following Scanorama or
344 Seurat with ICAT produces higher quality clusters compared to either of these integration
345 workflows alone.

346 By experimentally validating two uniquely predicted cell states in the embryo model
347 using single molecule FISH, we empirically demonstrated that the capability of ICAT to
348 accurately identify perturbed cell states surpasses current state-of-the-art integration workflows.
349 ICAT was the only method among the three tested that correctly identified these distinct states
350 and their perturbation responses, whereas both Seurat and Scanorama integration workflows
351 failed: highlighting ICAT as a substantial advancement in identifying and describing cell state
352 behaviors across biological conditions and perturbations.

353 Conclusion

354 One of the key promises of scRNA-seq technology is to more accurately define complex
355 biological systems by offering a more finely resolved perspective on transcriptional landscapes
356 than was previously possible. However, due to technical and biological noise, taking advantage
357 of such a powerful tool in diverse biological conditions has proven to be challenging. We
358 therefore developed the ICAT algorithm to improve the accuracy of cell state identification from
359 scRNA-seq data that includes perturbation treatments. By combining sparse feature weighting
360 with semi-supervised Louvain community detection, ICAT not only correctly identifies shared
361 and disparate cell states, but also produces interpretable results that aid in understanding
362 analytical outcomes and selecting the most suitable downstream analyses.

363 With the growing ubiquity of scRNA-seq experiments and the development of ever more
364 sophisticated experimental designs, we anticipate that ICAT will play an important role in highly
365 resolving cell state-specific responses to treatments and differing biological conditions. In the
366 future, the ICAT framework can easily be extended to operate on other single-cell modalities,
367 such as scATAC-seq, and will offer a potential route to intelligently cluster multi-modal data from
368 heterogeneous conditions.

Declarations

Availability of data and materials.

ICAT, *ncfs*, and *sslouvain* Python packages are freely available on pip and github (github.com/BradhamLab). Scripts to perform simulated data generation and benchmark comparisons are available as a SnakeMake workflow at (github.com/BradhamLab/ICAT_manuscript). Archived data files for both simulated and real data are also available at the same repository. The pipeline to align and quantify SMART-seq2 reads is freely available at (github.com/BradhamLab/scPipe), and the raw sequencing data is available at GEO accession GSE164240. Likewise, all scripts to segment PMCs and quantify HCR FISH signal are available at (github.com/BradhamLab/hcr_quantify) and the image data is available upon request.

Acknowledgements

We thank Prof. Masanao Yajima (Dept. of Mathematics and Statistics, Boston University) for helpful discussions. We thank Prof. Charles Ettensohn (Carnegie Mellon University, PA, USA) for the PMC antibody. We thank Jose Ordovas-Montanes and Sanjay Prakadan of the Shalek lab (MIT) for help with scRNA-seq library preparation. We thank Dr. Todd Blute for his advice and help with FACS and microscopy imaging.

Competing interests

The authors declare no competing interests.

Funding

This work was supported by NSF IOS 1656752 (CAB).

Author Contributions

The study was conceived by CAB and DYH. The computational work was performed by DYH, with advice from WEJ. JH and DTZ collected the urchin scRNA-seq data. NRS, KRM, and ATL performed the imaging experiments. DYH, KRM, and AG trained the PMC prediction model. The manuscript was written by DYH, CAB, DTZ and NRS, and edited by all coauthors.

Methods

Initial clustering

Prior to feature weighting, ICAT identifies initial clusters using Louvain community detection via the Scanpy Python package [15,33]. The neighbor graph is built using default parameters of *scanpy.pp.neighbors* with the exception of *n_neighbors* and *resolution* parameters, which are optimized for each dataset. Once the neighbor graph is constructed, ICAT performs Louvain community detection using the *scanpy.tl.louvain* function.

Neighborhood Component Feature Selection

Neighborhood Component Feature Selection (NCFS) was implemented in Python following the methods outlined in the original publication [16]. The implementation takes advantage of the just-in-time compiler library, NUMBA, to produce highly performant code [34]. NCFS optimizes feature weights to maximize the score function:

$$\xi(\mathbf{w}) = \sum_i \sum_j y_{ij} p_{ij} - \lambda \sum_{l=1}^{d} w_l^2$$

where $\mathbf{w}$ is the weight vector; $y_{ij} = 1$ if samples i and j belong to the same class, and is 0 otherwise; p_{ij} is the probability of selecting j as a reference to i in a kNN classifier; and $\lambda > 0$ is a regularization parameter. We alter the score function by including a class weight term, c_i, for each sample to account for class imbalance:

$$\xi(\mathbf{w}) = \sum_i c_i \sum_j y_{ij} p_{ij} - \lambda \sum_{l=1}^{d} w_l^2$$

Sample weights are calculated using methods outlined in the scikit-learn Python package [35,36]. Gradient ascent is then performed using the line method described in the original publication.

418 **Semi-supervised Louvain community detection**

419 To identify partitions in a graph, traditional Louvain community detection initializes each
420 node into singleton clusters. Nodes are then merged into other communities if doing so would
421 optimize a specified quality score [15]. To perform semi-supervsied clustering, ICAT initializes
422 communities with previously identified control cluster labels. Nodes representing perturbed cells
423 are placed into their own singleton clusters. During optimization, initialized control labels are set
424 immutable, while labels for perturbed cells are free to change. By keeping control labels
425 immutable, we simultaneously allow perturbed cells to merge into control communities — to
426 identify mixed cell states — while also allowing isolated nodes to form treatment-specific
427 communities. By default, ICAT optimizes graph partitions using the Reichardt-Burnholdt metric
428 [37]. Semi-supervised Louvain was implemented by extending the popular louvain-igraph
429 package (github.com/vtraag/louvain-igraph) to allow for immutable nodes.

430 **Subsampling by submodular optimization**

431 Due to the computational intensity of NCFS feature weighting, for large datasets it is
432 beneficial to learn feature weights on a subset of the overall cells. To select cells that correctly
433 capture the dataset variation, ICAT employs submodular optimization via the *Apricot* Python
434 package [22]. By default, ICAT uses the *facilityLocation* model in *Apricot* to successively select
435 cells that represent underrepresented regions of the data. The number of cells to select is
436 chosen by the user; this was set to 1,500 for both Kagohara and Kang datasets.

437 **Simulating scRNA-seq perturbation experiment data**

438 Simulated data was generated using a zero-inflated negative binomial distribution to
439 model gene counts. Our approach follows methods outlined by Büttner and colleagues [14],
440 used to simulate batch effects in scRNA-seq data, adopted to provide for multiple cell
441 populations as well as different cell identity resolutions and perturbation severity.

442 For each dataset, we simulated 1,500 genes. Letting $G = \{g_1, \ldots, g_{1,500}\}$ be the set of
443 all genes, to model M distinct cell identities, we simulate K marker genes such that

$G_k = \{g^1, \ldots, g^K\} \subset G$ and $K \sim B(n = 1500, p = p_{marker})$, where p_{marker} is set for each simulation. For simplicity, marker genes were made to be mutually exclusive, such that a gene could not be selected as a marker for multiple populations. Count matrices were constructed as follows:

Denote each cell identity y^k, where $Y = \{y^1, \ldots, y^M\}$ is the set of all identity labels. Assuming N cells, let y_i be the cell identity label for cell i such that $y_i \in Y$ and G_{y_i} is the set of marker genes associated with the given identity. For control cells, to generate the count matrix X, we simulate the cell-gene expression values $X_{ij} \sim NB(\mu_{ij}, r = r_j | p_{ij})$, where $r_j \sim U(1, 4)$ is the gene-wide dispersion factor and p_{ij} is the cell-gene specific dropout probability. Prior to marker gene application, gene-wide expected values are simulated using a Beta distribution, where $\mu_j \sim Beta(a = 2, b = 5) \cdot c$, where $c = 100$ is a scalar constant. For marker genes, this scalar c is modified to a cell-identity specific value where $c_{y_i} \sim \Gamma(a = 3, b = 3) \cdot 100$. Thus, cell-gene expected values are modeled:

$$\mu_{i,j} = \begin{cases} \beta(a = 2, b = 5) \cdot 100, & g_j \notin G_l \\ \beta(a = 2, b = 5) \cdot c_{y_i}, & g_j \in G_l \end{cases}$$

Letting $\mu_{y_i} = \{\mu_{y_i,1}, \ldots, \mu_{y_i,M}\}$ be the set expected values for all gene expression values for cells in population y_i, the cell-gene specific dropout probability, p_{ij} is set using the sigmoid function $p_{ij} = sigm\left(-\beta_0 + \beta_{y_i j} \cdot \mu_{ij}\right)$, where $\beta_0 = -1.5$ and $\beta_{y_i j} = median(\mu_{y_i})^{-1}$.

To simulate the count matrix for perturbed cells, X^{pert}, we use the general framework described above, but apply a scalar shift to gene-wide expected values for perturbed genes. First, genes to perturb are randomly sampled from G without replacement, such that $G^{pert} \subset G^{sim}$ is the set of all genes to perturb, and G^{sim} is the set of all allowable genes to perturb. By default, G^{sim} is the set of all non-marker genes such that $G^{sim} = G \setminus \bigcup_{k=1}^{k=M} G_k$. When

simulating a stimulated population, we specifically included all marker genes for target population y_{stim} in G^{pert} where the rest of the genes were again sampled from G^{sim}. Assuming gene $g_j \in G^{pert}$, a perturbation scalar is sampled from a Gamma distribution such that $s_j \sim \Gamma(a = 2, b = 2)$. To simulate expression changes due to perturbation, this scalar is multiplied by cell-gene expected values. Letting $\mu_{i,j}^{pert}$ be the expected value for gene g_j of cell i in X^{pert},

$$\mu_{i,j}^{pert} = \begin{cases} \mu_{ij}, & g_j \notin G^{pert} \\ \mu_{ij} \cdot s_j, & g_j \in G^{pert} \end{cases}$$

By plugging in μ_{ij}^{pert} into the described count simulation framework, we simulate both cell-gene dropout probabilities and the final counts. An exception is made for marker genes: in order to ensure expected population structure exists in both control and perturbed count matrices, dropout probabilities of marker genes were forced to be the same between the two conditions.

Simulating experimental conditions

To increase the severity of perturbation and cell identity separation, we varied $|G^{sim}|$ and p_{marker}, respectively. For perturbation severity experiments, $|G^{sim}|$ was set to 1%, 5%, 10%, 15%, 20%, or 25% of the total number of genes. To increase cell identity signal strength, p_{marker} was chosen from the range [0.007, 0.07] with a step size of 0.07. Given a binomial distribution, the average number of marker genes ranged from 10.5 - 105 genes per population.

Assessing performance in real datasets

Raw count matrices of publicly available datasets [4,7,21] were downloaded from GEO. All datasets were preprocessed using Scanpy, following typical workflows: first, cell counts were normalized by library size using the *scanpy.pp.normalize_total* function. Normalized counts were then log-transformed using the *scanpy.pp.log1p* function. Finally, treatment-specific highly variable genes were identified using the *scanpy.pp.highly_variable_genes* function, with

491 "flavor=seurat" setting, and with each dataset's treatment variable set as the batch key. The
492 union of highly variable genes was kept, and genes that were not highly variable in at least one
493 treatment set were removed.

494 **Tian processing**

495 The Tian CellMix count matrices and metadata files were downloaded from GEO with ID
496 GSE118767. To isolate more discrete populations, all cells not belonging to pure H2228,
497 HCC827, and H1975 cell mixtures, or the cell mixture with equal proportions, were filtered out.
498 Count matrices were preprocessed using *Scanpy* to remove genes expressed in less than three
499 cells, and the dataset of origin (originally the batch confounder) was set to the treatment
500 variable.

501 **Kagohara processing**

502 The Kagohara count matrices and metadata files were downloaded from GEO with ID
503 GSE137524. Genes present in less than 50 cells were removed, along with cells with reads for
504 less than 50 genes.

505 **Kang processing**

506 The Kang count matrices and metadata files were downloaded from GEO with ID
507 GSE96583. Cells described as doublets in the cell metadata files were removed from
508 downstream analysis. Further, genes present in less than 50 cells were removed, along with
509 cells with reads for less than 50 genes.

510 **Louvain parameter selection**

511 In Louvain community detection, two parameters k, the number of neighbors, and γ, a
512 resolution parameter, play important roles in identifying clusters. To optimize these parameters
513 for each dataset (both simulated and real), we filtered each unintegrated dataset to control cells,
514 and performed a grid search to find parameter pairs that optimized the Calinski-Harabarz score
515 as implemented in scikit-learn[21,28]. Neighbor values were searched on the interval [5, 50] with a

step size of 5. Likewise, we searched resolution values spanning [0.2, 1.25] with a step size of 0.05.

ICAT feature weighting

For the Kagohara and Kang datasets, feature weights were learned using only control cells (via the $ICAT_C$ implementation). Meanwhile, when analyzing all simulated datasets along with the Tian dataset, feature weights were jointly learned and subsequently collapsed across conditions using the $ICAT_{C+T}$ implementation. Genes were considered “informative” if they had a learned weight > 1. NCFS parameters used for each dataset are listed in Table S13.

Seurat integration

Seurat integration was performed using version 3.1 as outlined in the vignette (https://satijalab.org/seurat/articles/integration_introduction.html). Datasets were integrated over the treatment variable specified for each dataset. Default parameters were used with the exception of specifying calculation of 50 principal components, and explicitly passing k neighbors and γ resolution parameters to standardize parameters between methods. To follow the expected Seurat workflow as best as possible, clusters were identified using the Seurat function, FindClusters, with the "algorithm=1" to perform Louvain community detection.

Scanorama integration

Scanorama integration was performed using version 1.4, and passing AnnData objects to the *run_scanorama()* function. Default parameters were used, with the exception of k and γ parameters being specified for each dataset. After Scanorama integration, clusters were identified using the *scanpy.tl.louvain* function from Scanpy

Quality metrics

All metrics, with the exception of LISI, were calculated using the scikit-learn Python package. Metrics that rely on distance measures, Davies-Bouldin (DB) and LISI scores, were calculated using the euclidean distance over the first 50 principal components.

Davies-Bouldin

For DB, scores closer to zero represent a tighter cluster that is more separated from other clusters. To facilitate comparison between metrics for each dataset, the negative of DB scores was scaled between [0, 1] using the MinMaxScaler preprocessor from scikit-learn, where 1 now represents the best performing method.

LISI

LISI metrics were calculated using the "lisi" R package [19]. The LISI score naturally falls between 0 and the number of batches/treatments in the dataset (representing total separation and perfect mixing, respectively). Because we expect mixing in only shared cell identities, PCA was first run on the complete data matrix, then cells from non-overlapping cell identities were removed, and LISI scores were then calculated using PCs from shared identities. Because LISI calculates a score for each cell, cell scores were averaged for each dataset-method comparison. To facilitate comparison to other metrics, LISI scores were scaled between 0 and 1 using the formula:

$$LISI_{scaled} = \frac{LISI - 1}{P - 1}$$

where P is the number of treatment conditions in the dataset.

Population specific F1-scores

To assess each method's ability to detect stimulated and asymmetrical populations, we created population-specific F1-scores. For each population of interest, we identified the cluster where a plurality of cells was assigned, and denoted this cluster as c. For each cell within the population of interest, we created a binary vector, v, denoting whether the cell was assigned to cluster c.

$$v_i = \begin{cases} 1 & y_i = c \\ 0 & y_i \neq c \end{cases}$$

where y_i is the cluster label for cell i. Assuming perfect isolation where all cells belong to the same cluster, all entries would be 1. Creating a vector of ones: $\mathbf{u} = \vec{1}$, serves as the basis for

566 comparison. We passed v and u to the *sklearn.metrics.f1_score* function to calculate the
567 population-specific F1-score [35]. In the case of F1-Unique, where multiple populations of
568 interest exist, we calculate F1-scores for each population independently and average the scores
569 between populations. F1-Stim was calculated based solely on the activated population.

570 **Animals and perturbations**

571 Adult *Lytechinus variegatus* sea urchins were obtained from the Duke University Marine
572 Laboratory (Beaufort, NC) or from Reeftopia (Miami, FL). Gamete harvesting and embryo
573 culturing was performed as described [30,38]. Embryos were treated with chlorate to inhibit the
574 formation of sulfated proteoglycans as described [27]. MK-886 (Tocris) was reconstituted in
575 DMSO, and dose–response experiments were performed to obtain the optimal working doses.
576 The optimal MK-886 concentration was determined to be 1.5-2 μM.

577 **PMC preparation and FACS for scRNA-seq analysis**

578 Embryos were disaggregated at 18 hours post-fertilization (hpf) according to McClay and
579 Fink [39], with modification as follows: Approximately 3000 embryos at 18 hpf were collected in
580 artificial seawater (ASW) supplemented with pronase (0.66 mg/mL, Sigma) and immediately
581 centrifuged for 1 minute at 800 x g, then resuspended in calcium-free seawater (CFSW: 450
582 mM sodium chloride, 9 mM potassium chloride, 48 mM magnesium sulfate, 6 mM sodium
583 bicarbonate) containing 20 mg/mL BSA, followed by the addition of 1 volume of hyaline
584 extraction medium (HEM) [39] and allowed to incubate for 4 minutes at room temperature (RT).
585 Embryos were then centrifuged for 1 minute at 800 x g, resuspended in CFSW [39], triturated
586 with decreasing diameter pipettes for 30 seconds total, then centrifuged at 1000 x g for 2
587 minutes to pellet the dissociated cells. Single cells were then resuspended in CFSW,
588 centrifuged for 1 minute, then incubated in ASW containing 1 mg/mL BSA with 6a9 anti-PMC
589 primary antibody (1:5) and anti-mouse Alexa-Fluor 488 labeled secondary antibody (1:500) for 5
590 minutes. Following washes, cells were suspended in CFSW containing Sytox Blue Dead Cell
591 Stain (1:1000, ThermoFisher) to label dead cells for exclusion, then sorted using a Sony SH800

592 Cell Sorter. Cells were gated to exclude doublets and dead cells and collected into 96-well
593 plates preloaded with RNA lysis buffer at one cell per well, for processing for single cell library
594 preparation.

595 **scRNAseq library preparation and sequencing**

596 RNA quantitation and integrity were determined using a Qubit® 2.0 Fluorometer (Life
597 Technologies) and a 2100 Bioanalyzer (Agilent Technologies). scRNAseq libraries were
598 generated using Smart-seq2 on FACS-sorted PMCs as described [24,40]. Libraries were
599 sequenced on an Illumina NXT platform (BU Microarray and Sequencing Resource Core
600 Facility) using 75 bp paired end reads. We obtained a median of 1.4M reads per cell over a total
601 of 416 cells. 144 of these cells were untreated (ASW, artificial sea water), 66 were treated with
602 DMSO (vehicle), 206 with chlorate, and 192 with MK-886. For all reads, quality control was
603 performed prior to alignment with fastp [41] using default parameters. Roughly 97% of reads
604 passed quality control measures with a median of ~1.3M reads per cell. Quality-controlled reads
605 were aligned to the *L. variegatus* genome [42] using STAR [43], and alignments were quantified
606 using featureCounts [44]. A median of 148,142 reads per cell were mapped and assigned to
607 annotated regions within the genome. To ensure only well-sequenced cells and genes were
608 considered, we removed cells with less than 25,000 mapped reads, then genes with reads
609 assigned in less than five cells, using Scanpy [33]. Once the final count matrix was generated,
610 we normalized reads between cells and performed batch-correction using the SCRAN package
611 [45,46]. Assuming some non-PMCs escaped FACS, we evaluated the identity of both the sorted
612 and remainder cells from the FACS analysis for the expression of known marker genes for each
613 major cell type [47]. We then performed a final filtering step that removed sorted cells that lack
614 expression of canonical PMC marker genes. After filtering, 119 ASW cells, 26 DMSO-treated
615 cells, 154 chlorate-treated cells, and 134 MK-treated cells remained. For analysis, ASW and
616 DMSO vehicle controls were combined into a single control group since the number of DMSO
617 cells were too small to consider them as a representative sample for these comparisons and

618 since comparisons of DMSO and ASW cells did not reveal significant expression differences.
619 Prior to cluster analysis, genes were subsetted to highly variable genes identified in control cells
620 using the *highly_variable_genes* function in *Scanpy*. For clustering, neighbor parameter k=15
621 and resolution parameter γ = 1 were used for both ICAT clustering, and subsequent Louvain
622 Community Detection in Seurat and Scanorama workflows.

623 **Assessing treatment effect on cell state composition**

624 To assess whether cell state membership was dependent upon treatment, we performed
625 G-tests on each cell state using the scipy.stats.*power_divergence* function in the *scipy* Python
626 package [48,49]. P-values were then adjusted using the *statsmodels.multitest.multipletests*
627 function from *statsmodel* in Python, with *method="fdr_bh"* [50]. Associations were considered
628 significant with α = 0.05. Cell states with compositions dependent upon treatment status were
629 further evaluated with post-hoc pairwise Fisher's Exact test to determine which treatments
630 affected cell state membership compared to control. Tests were performed using
631 scipy.stats.*fisher_exact* function from scipy, with p-values again adjusted using the *multipletests*
632 function with the same parameters.

633 **Single Molecule FISH: probe sets, amplifiers and buffers**

634 Embryos were collected and fixed in 4 % paraformaldehyde at 18 hpf. Hybridization
635 chain reaction (HCR)-FISH was performed as described [31] using probes, hairpins and buffers
636 obtained from Molecular Instruments, Inc. Embryos were incubated with hairpins for 2.5 hours,
637 then signals were captured as 3-D z-stacks using confocal microscopy.

638 **Immunostaining and confocal microscopy**

639 Immunolabeling was performed as described [30,51,52]. PMCs were detected using
640 primary antibody 6a9 (1:50; from Prof. Charles Ettensohn, Carnegie Mellon University,
641 Pittsburgh, PA, USA) and secondary goat-anti mouse antibody labeled with Alexa 546 (1:500;
642 ThermoFisher). Hoechst 33342 (1:1000; Thermo Fisher) was included as a nuclear

643 counterstain. Confocal microscopy was performed using a Nikon-C2 confocal & Ti-E Spectral
644 imaging system. Confocal z-stacks were projected using Napari [53].

645 **PMC segmentation**

646 To isolate individual PMCs from the 3-D image stack, a random forest PMC classifier
647 was trained using the “pixel classification” workflow from Ilastik [32]. The model was trained
648 using DMSO and MK-treated embryos (n=70) to produce “PMC” and “Background” probabilities.
649 Prior to training, images were Z-cropped to include only z-slices with PMC signal. Each slice
650 was then contrast and intensity adjusted using the functions *exposure.adapt_hist* and
651 *exposure.rescale_intensity* functions from *scikit-image (skimage)* [54] library in Python. PMC
652 probabilities were exported from Ilastik for post-processing and PMC segmentation by a custom
653 Python script.

654 Briefly, exported Ilastik probabilities were first segmented using
655 *filters.apply_hysteresis_threshold* from *skimage* with parameters “low”=0.45 and “high”=0.5. The
656 resulting binary mask was then refined via morphological opening using
657 *morphology.binary_opening*, again from *skimage*. The opened mask was subsequently used as
658 seed points for watershed segmentation (*skimage.segmentation.watershed*) using the image
659 gradients from the original data (*skimage.filters.sobel, axis=0)*. To restrict PMC labels to typical
660 PMC sizes and to better separate dense PMC areas, individual regions with areas exceeding
661 600 voxels were further segmented using a stricter threshold workflow: first, PMC probabilities
662 were smoothed using a gaussian filter (*skimage.filters.gaussian*). Second, starting with
663 threshold t = 0.8, a binary mask of the smoothed probabilities was generated by finding all
664 voxels > t and performing a morphological opening. Isolated objects were then detected using
665 *measure.label* from *skimage.* If the number of identified objects did not exceed one, the process
666 was repeated after incrementing t by 0.025 until the original region was split. The split labels
667 were then filled back into the original larger region using watershed. To split cell segmentations
668 with unrealistic 2-D diameters, this process was repeated for any labeled region having a

669 diameter greater than 15 voxels as well as an area greater than 500 voxels. To correctly
670 segment vertically stacked PMCs, any label that spanned more than seven z-slices was split
671 after calculating the label diameter at each slice. Given that PMCs are spherical, it was
672 reasoned local minima in diameter would indicate the end of a cell, and thus these served as
673 break points between labels. Finally, to remove unrealistically small labels, the final
674 segmentation was produced by removing any label that occupied less than 55 voxels.

675 **HCR FISH quantification and analysis**

676 Prior to quantification, images were filtered to remove embryos that displayed either poor
677 contrast or failed to be entirely captured by the z stack. After quality control, 26 DMSO and 37
678 MK treated embryos were retained. IIndividual PMCs were then labeled using the previously
679 described methodology. After label generation, *pks2* and *sm50* HCR FISH image channels were
680 preprocessed using FISH-quant v2 [55] to scale and normalize intensities and to perform
681 background subtraction and autofluorescence removal. Gene expression was quantified for
682 each predicted PMC by calculating the average signal intensity as described [31]. Prior to
683 analysis, PMC-specific intensities were standardized to a [0, 1] scale to allow for inter-embryo
684 comparisons. Cells were considered to have decreased (-) or increased (+) expression for a
685 given gene if its standardized expression fell below the 25th or above the 75th percentile,
686 respectively. We then calculated the proportion of *sm50+/pks2-* and *sm50-/pks2+* PMCs for
687 each embryo. Statistical analysis was performed using a general linear model to perform
688 binomial logistic regression to assess the effect of treatment status on phenotype proportion
689 while also accounting for the fertilization batch. The model was fit using the *statsmodels* [50]
690 Python package using the formula *phenotypic* + *not_phenotypic* ~ *treatment* + *fertilization*,
691 where *phenotypic* is the number of cells displaying the phenotype of interest, *not_phenotypic* is
692 the number of cells not displaying the phenotype of interest, *treatment* is treatment status, and
693 *fertilization* is the fertilization batch for the given embryo.
694

695

696

1. Schelker M, Feau S, Du J, Ranu N, Klipp E, MacBeath G, et al. Estimation of immune cell content in tumour tissue using single-cell RNA-seq data. Nat Commun. 2017 Dec 11;8(1):2032.
2. Wagner DE, Weinreb C, Collins ZM, Briggs JA, Megason SG, Klein AM. Single-cell mapping of gene expression landscapes and lineage in the zebrafish embryo. Science. 2018 Jun 1;360(6392):981–7.
3. Luecken MD, Theis FJ. Current best practices in single-cell RNA-seq analysis: a tutorial. Mol Syst Biol. 2019 Jun 1;15(6):e8746.
4. Kang HM, Subramaniam M, Targ S, Nguyen M, Maliskova L, McCarthy E, et al. Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. Nat Biotechnol. 2017 Dec 11;36(1):89–94.
5. Haber AL, Biton M, Rogel N, Herbst RH, Shekhar K, Smillie C, et al. A single-cell survey of the small intestinal epithelium. Nature. 2017 Nov;551(7680):333–9.
6. Lähnemann D, Köster J, Szczurek E, McCarthy DJ, Hicks SC, Robinson MD, et al. Eleven grand challenges in single-cell data science. Genome Biol. 2020 Feb 7;21(1):31.
7. Kagohara LT, Zamuner F, Davis-Marcisak EF, Sharma G, Considine M, Allen J, et al. Integrated single-cell and bulk gene expression and ATAC-seq reveals heterogeneity and early changes in pathways associated with resistance to cetuximab in HNSCC-sensitive cell lines. Br J Cancer. 2020 Jul;123(1):101–13.
8. Regev A, Teichmann SA, Lander ES, Amit I, Benoist C, Birney E, et al. The Human Cell Atlas. eLife. 2017 Dec 5;6.
9. Schaum N, Karkanias J, Neff NF, May AP, Quake SR, Wyss-Coray T, et al. Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. Nature. 2018 Oct;562(7727):367–72.
10. Perillo M, Oulhen N, Foster S, Spurrell M, Calestani C, Wessel G. Regulation of dynamic pigment cell states at single-cell resolution. Newmark PA, Bronner ME, Rast JP, editors. eLife. 2020 Aug 19;9:e60388.
11. Stuart T, Butler A, Hoffman P, Hafemeister C, Papalexi E, Mauck WM, et al. Comprehensive Integration of Single-Cell Data. Cell. 2019 Jun 13;177(7):1888-1902.e21.
12. Hie B, Bryson B, Berger B. Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. Nat Biotechnol. 2019 Jun 6;37(6):685–91.
13. Luecken MD, Büttner M, Chaichoompu K, Danese A, Interlandi M, Mueller MF, et al. Benchmarking atlas-level data integration in single-cell genomics. Nat Methods. 2022 Jan;19(1):41–50.
14. Büttner M, Miao Z, Wolf FA, Teichmann SA, Theis FJ. A test metric for assessing single-cell RNA-seq batch correction. Nat Methods. 2019 Jan 20;16(1):43–9.
15. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. J Stat Mech Theory Exp. 2008 Oct 9;2008(10):P10008.
16. Yang W, Wang K, Zuo W. Neighborhood Component Feature Selection for High-Dimensional Data. J Comput [Internet]. 2012 Jan 1 [cited 2018 Nov 12];7(1). Available from: http://ojs.academypublisher.com/index.php/jcp/article/view/5076
17. Traag V. vtraag/louvain-igraph [Internet]. 2021 [cited 2021 Mar 15]. Available from: https://github.com/vtraag/louvain-igraph
18. Hubert L, Arabie P. Comparing partitions. J Classif. 1985 Dec 1;2(1):193–218.
19. Korsunsky I, Millard N, Fan J, Slowikowski K, Zhang F, Wei K, et al. Fast, sensitive and accurate integration of single-cell data with Harmony. Nat Methods. 2019 Dec;16(12):1289–96.
20. Satija R, Farrel JA, Gennert D, Schier AF, Regev A. Spatial reconstruction of single-cell gene expression data. Nat Biotechnol. 2015;33(5).
21. Tian L, Dong X, Freytag S, Lê Cao KA, Su S, JalalAbadi A, et al. Benchmarking single cell RNA-sequencing analysis pipelines using mixture control experiments. Nat Methods. 2019

Jun 27;16(6):479–87.
22. Schreiber J, Bilmes J, Noble WS. apricot: Submodular selection for data summarization in Python. J Mach Learn Res. 2020;21(161):1–6.
23. Davies DL, Bouldin DW. A Cluster Separation Measure. IEEE Trans Pattern Anal Mach Intell. 1979 Apr;PAMI-1(2):224–7.
24. Picelli S, Faridani OR, Björklund ÅK, Winberg G, Sagasser S, Sandberg R. Full-length RNA-seq from single cells using Smart-seq2. Nat Protoc. 2014 Jan 2;9(1):171–81.
25. Pidgeon GP, Lysaght J, Krishnamoorthy S, Reynolds JV, O'Byrne K, Nie D, et al. Lipoxygenase metabolism: roles in tumor progression and survival. Cancer Metastasis Rev. 2007 Oct 18;26(3):503.
26. Simionato GB, da Silva ACR, Oliva AH de, de Abreu Furquim EM, Couto MCR, Shinohara AL, et al. Lack of 5-lipoxygenase in intramembranous and endochondral 129Sv mice skeleton and intramembranous healing. Arch Oral Biol. 2021 Nov 1;131:105266.
27. Piacentino ML, Zuch DT, Fishman J, Rose S, Speranza EE, Li C, et al. RNA-Seq identifies SPGs as a ventral skeletal patterning cue in sea urchins. Dev Camb Engl. 2016 Feb 15;143(4):703–14.
28. Lyons DC, Martik ML, Saunders LR, McClay DR. Specification to biomineralization: following a single cell type as it constructs a skeleton. Integr Comp Biol. 2014 Oct;54(4):723–33.
29. Sun Z, Ettensohn CA. Signal-dependent regulation of the sea urchin skeletogenic gene regulatory network. Gene Expr Patterns. 2014;16:93–103.
30. Piacentino ML, Ramachandran J, Bradham CA. Late Alk4/5/7 signaling is required for anterior skeletal patterning in sea urchin embryos. Development. 2015 Mar 1;142(5):943–52.
31. Choi HMT, Schwarzkopf M, Fornace ME, Acharya A, Artavanis G, Stegmaier J, et al. Third-generation in situ hybridization chain reaction: multiplexed, quantitative, sensitive, versatile, robust. Development. 2018 Jun 26;145(12):dev165753.
32. Berg S, Kutra D, Kroeger T, Straehle CN, Kausler BX, Haubold C, et al. ilastik: interactive machine learning for (bio)image analysis. Nat Methods 2019. 2019 Sep 30;1–7.
33. Wolf FA, Angerer P, Theis FJ. SCANPY: large-scale single-cell gene expression data analysis. Genome Biol. 2018 Dec 6;19(1):15.
34. Lam SK, Pitrou A, Seibert S. Numba: a LLVM-based Python JIT compiler. In: Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC [Internet]. New York, NY, USA: Association for Computing Machinery; 2015 [cited 2021 Mar 3]. p. 1–6. (LLVM '15). Available from: https://doi.org/10.1145/2833157.2833162
35. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. J Mach Learn Res. 2011;12(85):2825–30.
36. King G, Zeng L. Logistic Regression in Rare Events Data. Polit Anal. 2001;9:137–63.
37. Reichardt J, Bornholdt S. Statistical mechanics of community detection. Phys Rev E. 2006 Jul 18;74(1):016110.
38. Bradham CA, McClay DR. p38 MAPK is essential for secondary axis specification and patterning in sea urchin embryos. Development. 2006 Jan 1;133(1):21–32.
39. McClay DR, Fink RD. Sea urchin hyalin: Appearance and function in development. Dev Biol. 1982 Aug 1;92(2):285–93.
40. Trombetta JJ, Gennert D, Lu D, Satija R, Shalek AK, Regev A. Preparation of Single-Cell RNA-Seq Libraries for Next Generation Sequencing. Curr Protoc Mol Biol. 2014 Jul 1;107:4.22.1-17.
41. Chen S, Zhou Y, Chen Y, Gu J. Fastp: An ultra-fast all-in-one FASTQ preprocessor. Bioinformatics. 2018;34(17):i884–90.
42. Li Y, Omori A, Flores RL, Satterfield S, Nguyen C, Ota T, et al. Genomic insights of body plan transitions from bilateral to pentameral symmetry in Echinoderms. Commun Biol.

2020 Jul 10;3(1):1–10.
43. Dobin A, Davis CA, Schlesinger F, Drenkow J, Zaleski C, Jha S, et al. STAR: ultrafast universal RNA-seq aligner. Bioinformatics. 2013 Jan 1;29(1):15–21.
44. Liao Y, Smyth GK, Shi W. featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. Bioinformatics. 2014 Apr 1;30(7):923–30.
45. L. Lun AT, Bach K, Marioni JC. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. Genome Biol. 2016 Dec 27;17(1):75.
46. Haghverdi L, Lun ATL, Morgan MD, Marioni JC. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. Nat Biotechnol. 2018 Apr 2;36(5):421–7.
47. Hogan JD, Keenan JL, Luo L, Ibn-Salem J, Lamba A, Schatzberg D, et al. The developmental transcriptome for Lytechinus variegatus exhibits temporally punctuated gene expression changes. Dev Biol. 2020 Apr 15;460(2):139–54.
48. Sokal R, Rohlf F. Biometry : the principles and practice of statistics in biological research / Robert R. Sokal and F. James Rohlf. SERBIULA Sist Libr 20. 2013 Apr;
49. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nat Methods. 2020;17:261–72.
50. Seabold S, Perktold J. statsmodels: Econometric and statistical modeling with python. In: 9th Python in Science Conference. 2010.
51. Bradham CA, McClay DR. p38 MAPK is essential for secondary axis specification and patterning in sea urchin embryos. Dev Camb Engl. 2006 Jan;133(1):21–32.
52. Gross JM, Peterson RE, Wu SY, McClay DR. LvTbx2/3: a T-box family transcription factor involved in formation of the oral/aboral axis of the sea urchin embryo. Dev Camb Engl. 2003 May;130(9):1989–99.
53. Nicholas Sofroniew, Talley Lambert, Kira Evans, Juan Nunez-Iglesias, Kevin Yamauchi, Ahmet Can Solak, et al. napari/napari: 0.3.7rc3 [Internet]. Zenodo; 2020 [cited 2020 Sep 3]. Available from: https://zenodo.org/record/4012782#.X1EdIPEpCHs
54. van der Walt S, Schönberger JL, Nunez-Iglesias J, Boulogne F, Warner JD, Yager N, et al. scikit-image: image processing in Python. PeerJ. 2014 Jun;2:e453.
55. Imbert A, Ouyang W, Safieddine A, Coleno E, Zimmer C, Bertrand E, et al. FISH-quant v2: a scalable and modular analysis tool for smFISH image analysis [Internet]. 2021 Jul [cited 2021 Oct 25] p. 2021.07.20.453024. Available from: https://www.biorxiv.org/content/10.1101/2021.07.20.453024v1

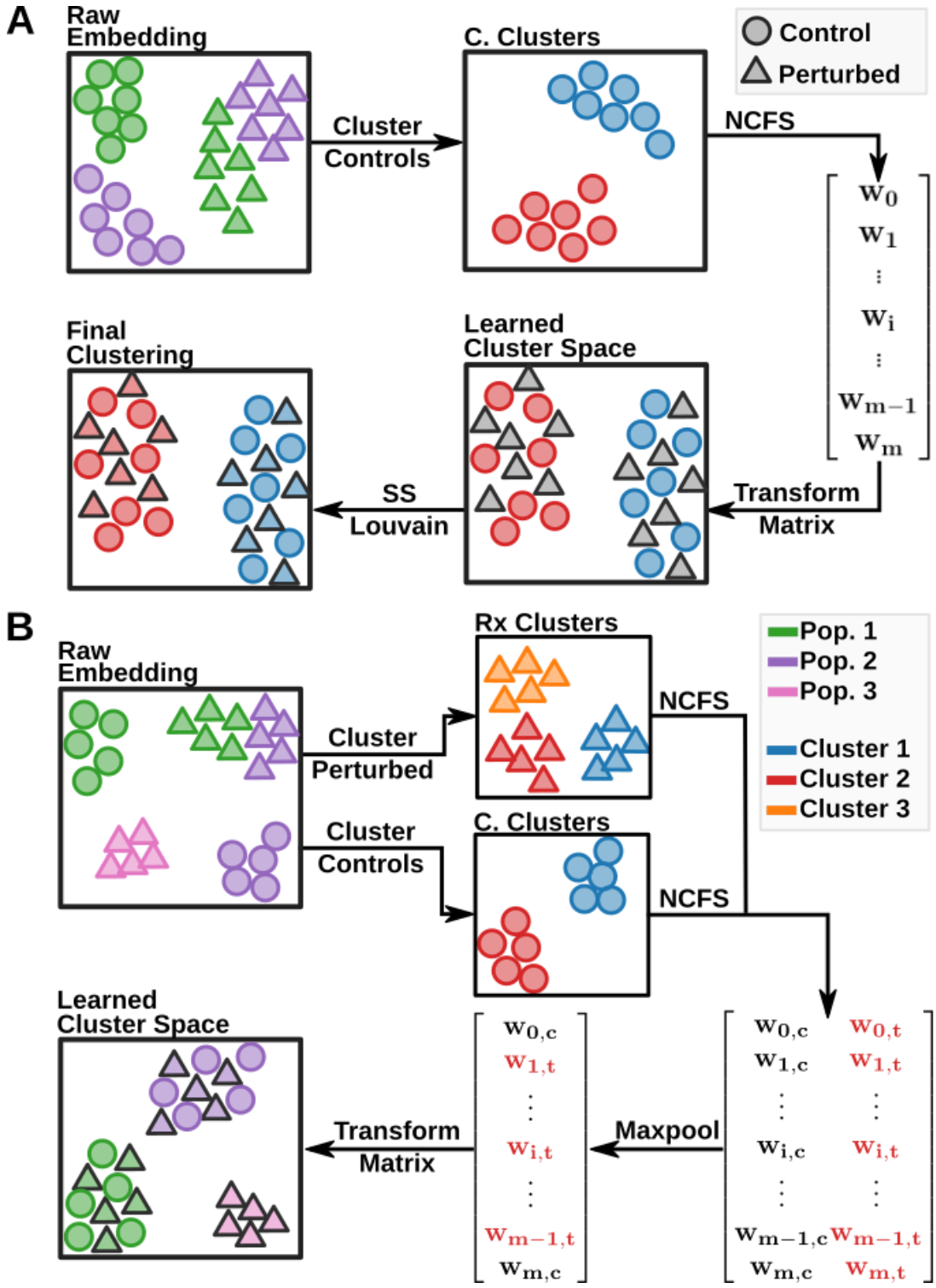
A
Raw Embedding
Cluster Controls
C. Clusters
Control
Perturbed
NCFS
w_0
w_1
w_i
w_{m-1}
w_m
Final Clustering
SS Louvain
Learned Cluster Space
Transform Matrix
B
Raw Embedding
Cluster Perturbed
Rx Clusters
NCFS
Pop. 1
Pop. 2
Pop. 3
Cluster 1
Cluster 2
Cluster 3
Cluster Controls
C. Clusters
NCFS
Learned Cluster Space
Transform Matrix
$w_{0,c}$
$w_{1,t}$
$w_{i,t}$
$w_{m-1,t}$
$w_{m,c}$
Maxpool
$w_{0,c}$ $w_{0,t}$
$w_{1,c}$ $w_{1,t}$
$w_{i,c}$ $w_{i,t}$
$w_{m-1,c}$ $w_{m-1,t}$
$w_{m,c}$ $w_{m,t}$

Figure 1: Overview of the ICAT algorithm. A. The schematic illustrates the $ICAT_C$ implementation of ICAT. To identify cell states across treatments, ICAT first performs self-supervised feature weighting to find genes that discriminate cell identities among control cells alone, followed by semi-supervised clustering using the newly transformed expression matrix. To learn feature weights, ICAT clusters control cells using Louvain community detection, then uses the newly generated cluster labels as input into NCFS to weight genes by their predictiveness. After applying the learned gene weights to the original gene expression matrix, ICAT clusters both treated and control cells together using a semi-supervised implementation of Louvain community detection. During this process, ICAT holds the previously identified cluster labels for the control cells immutable. **B.** The schematic illustrates the $ICAT_{C+T}$ implementation, which expands feature weighting to treated cells to identify asymmetrical populations between treatments. Cells are split along treatments and independently clustered using the Louvain method, then cluster labels are used to learn gene weights using NCFS in each treatment set independently. To retain asymmetrically informative genes, weights for each treatment are concatenated row-wise and subsequently reduced to the maximum weight using a row-wise maxpool. The reduced weight vector is then used to transform the original count matrix.

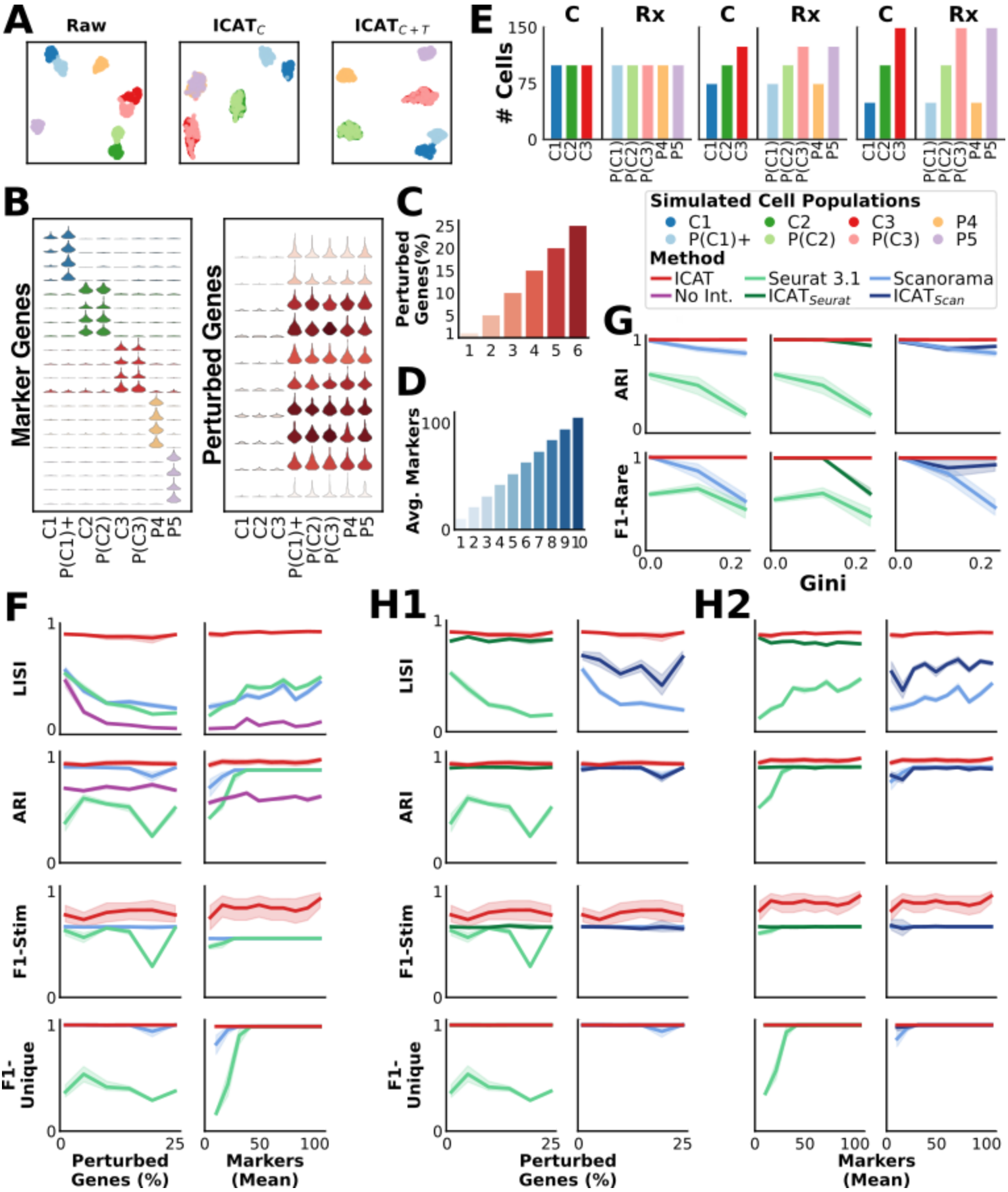
A
Raw
ICAT$_C$
ICAT$_{C+T}$
B
Marker Genes
Perturbed Genes
C1
P(C1)+
C2
P(C2)
C3
P(C3)
P4
P5
C
Perturbed Genes(%)
D
Avg. Markers
E
C
Rx
Cells
Simulated Cell Populations
C1
P(C1)+
C2
P(C2)
C3
P(C3)
P4
P5
Method
ICAT
No Int.
Seurat 3.1
ICAT$_{Seurat}$
Scanorama
ICAT$_{Scan}$
G
ARI
F1-Rare
Gini
F
H1
H2
LISI
ARI
F1-Stim
F1-Unique
Perturbed Genes (%)
Markers (Mean)

Figure 2. ICAT outperforms existing methods for cell state identification and is robust to experimental conditions. A. UMAP projections of raw (left), $ICAT_C$ processed (middle), and $ICAT_{C+T}$ processed count matrices for the simulated data. Projections show $ICAT_C$ and $ICAT_{C+T}$ correctly mix shared populations (red and green), while only $ICAT_{C+T}$ isolates asymmetrical populations (purple, yellow). ICAT performance for simulated data was further evaluated using the $ICAT_{C+T}$ implementation only. **B.** Select marker and perturbed gene expression patterns are displayed as violin plots for three simulated control cell types (1-3) under normal (C) and perturbed (P) conditions. P(C1)+ is a stimulated and perturbed version of cell type 1; perturbation-specific cell types P4 and P5 express distinct marker genes. **C.** The percentage of perturbed genes used to assess robustness to perturbation severity is shown; values range from 1-25%. **D.** The average number of marker genes per cell identity used to assess robustness to signal strength is shown; values range from 10 to 105 (0.7 - 7% of total genes). **E.** The set of cell identity proportions used to test the ability to identify rare cell states is shown. The number of cells per treatment-label pair ranges from 50 to 175. **F.** Method performance is compared as the fraction of perturbed genes increases (left), and as the average number of marker genes per population increases (right). Results are depicted as averages with 95% confidence intervals shown by shading. **G.** Method performance is compared as the proportion of cell types is varied (left). The Gini coefficient reflects the degree of population size disproportion among cell states. **H.** The performance of integration methods alone or in combination with ICAT (as $ICAT_{Seurat}$ or $ICAT_{Scan}$) is compared across the same metrics for the perturbation (H1) and signal strength experiments (H2).

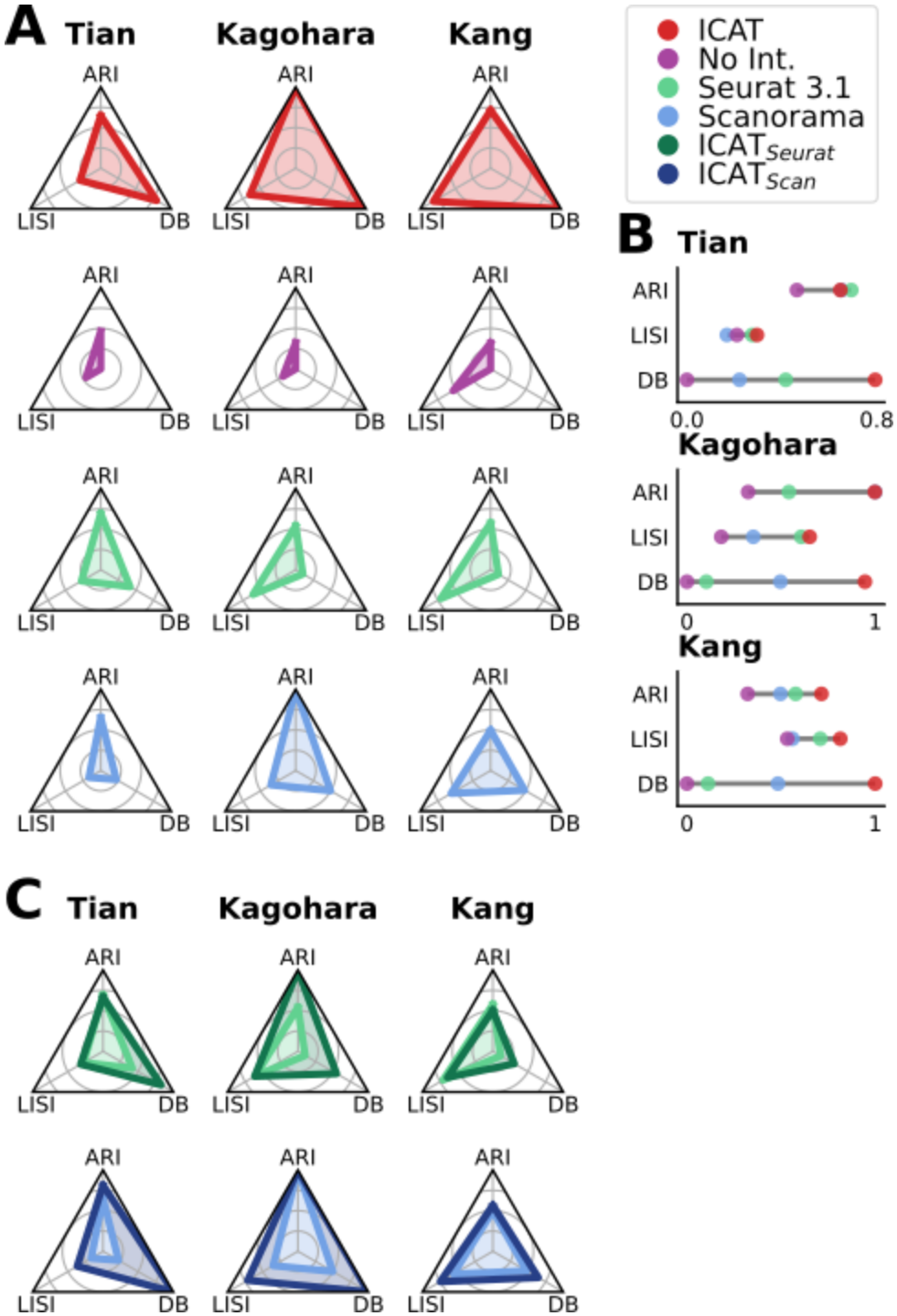

Figure 3: ICAT outperforms existing integration methods in identifying cell states across treatments in real datasets. A. Spider plots compare the performance for each algorithm within each dataset for ARI, LISI, and DB quality metrics. **B.** Lollipop plots highlight the differences in the metrics for each method across the same datasets. **C.** Spider plots comparing Seurat to $ICAT_{Seurat}$ and Scanorama to $ICAT_{Scan}$, respectively. All metrics are scaled from 0 to 1, where 1 is best (closest to the apex of each corner). Abbreviations: DB Davies-Bouldin metric, and otherwise as in Fig. 2. ICAT performance for the Tian dataset was evaluated using the $ICAT_{C+T}$ implementation, while Kagohara and Kang datasets were evaluated using the $ICAT_{C}$ implementation.

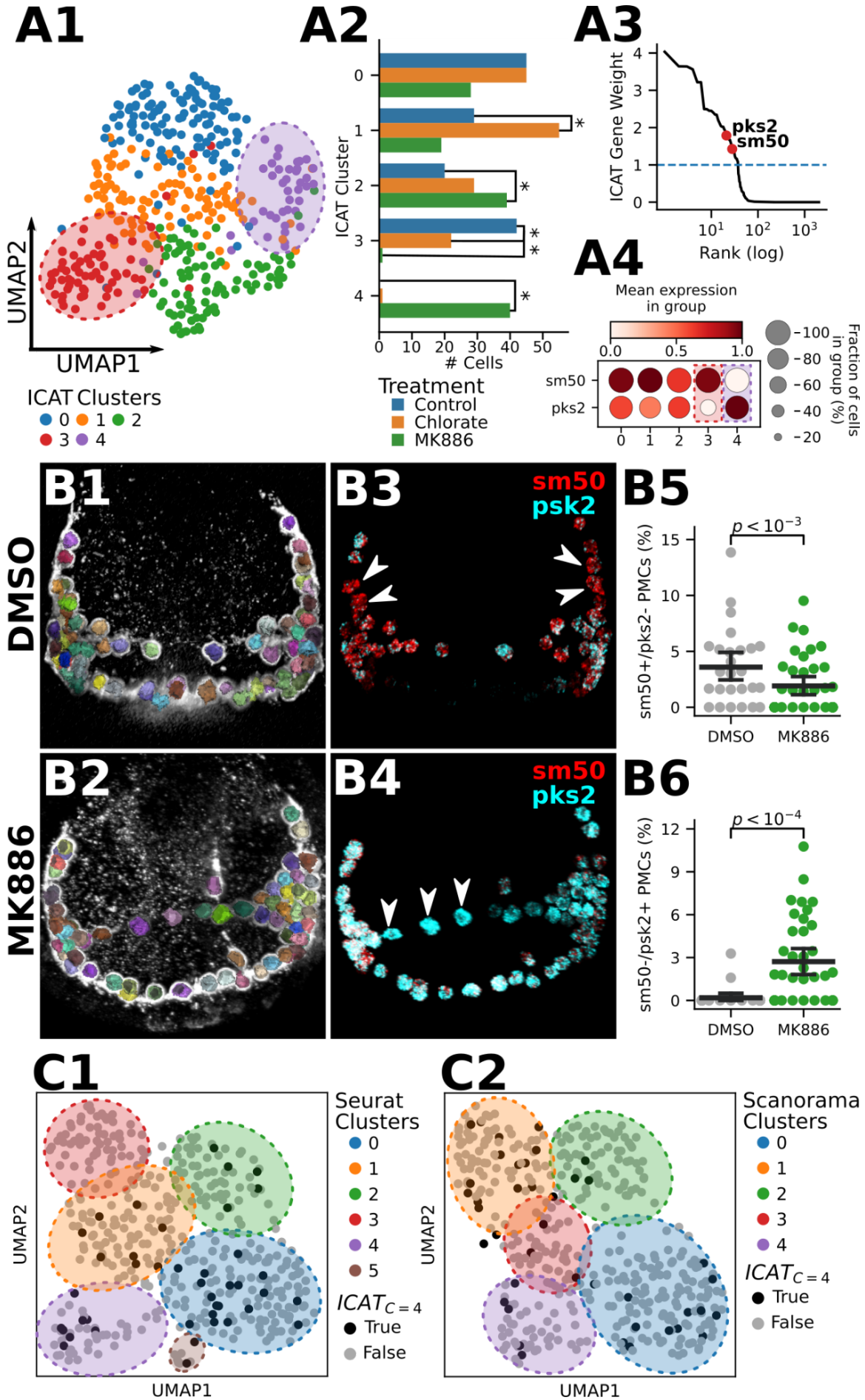

Figure 4: ICAT accurately characterizes subpopulation response to perturbation. SMART-seq2 was performed on cells isolated from controls or from embryos treated with perturbants chlorate or MK-886. **A.** ICAT predicts five clusters from the combined data (A1), with treatment-dependent subpopulation compositions (pairwise Fisher's Exact

test, fdr < 0.05) (A2). ICAT-calculated gene weights plotted by rank show *sm50* and *pks2* as highly informative for cell-state membership, with 37 total genes being considered informative (weight > 1) (A3). MK-induced differences are defined by reciprocal expression patterns of ICAT-identified genes, *sm50* and *pks2,* in subpopulations 3 (sm50+/pks2-) and 4 (sm50-/pks2+) (A4). **B.** ICAT predictions for control (B1,3,5) and MK-866 (B2,4,6) are validated by HCR FISH analysis. **B1-2.** Automated detection and segmentation of PMCs *in vivo* from 3-D image data is shown as various colors; each color represents an individual cell. **B3-4.** The expression of *sm50* (red) and *pks2* (cyan) transcripts are shown in the same embryos; arrowheads indicate PMCs that express only *sm50* or only *pks2*. **B5-6.** MK-866 treated embryos exhibit statistically significantly fewer *sm50+/pks2-* PMCs (B5) and more *sm50-/pks2+* PMCs (B6) than controls, consistent with the predictions from ICAT (A). Dot plot error bars denote 95% confidence intervals of the mean with each dot representing an individual embryo ($n_{DMSO} = 26$, $n_{MK886} = 37$; B5 (Binomial GLM, μ_{DMSO}= 3.59%, μ_{MK886}= 1.91%; β_{MK886}= -0.71, $p < 10^{-3}$; B6 Binomial GLM, μ_{DMSO}= 0.19%, μ_{MK886}= 2.71%; β_{MK886}= 2.60, $p < 10^{-4}$). **C.** Integration methods fail to identify the MK886-enriched *sm50+/pks2-* subpopulation (ICAT cluster 4). UMAP projections show clusters generated by Seurat (C1) or Scanorama (C2) for the same data, showing with the cells comprising the MK866-specific asymmetric *sm50+/pks2-* population (ICAT cluster 4, black dots) distributed across the multiple clusters generated by each method. The clusters produced by each integration method are indicated by colored regions.