Bioinformatics, 37(14), 2021, 1972–1981
doi: 10.1093/bioinformatics/btab038
Advance Access Publication Date: 28 January 2021
Original Paper

Gene expression

Using single-cell cytometry to illustrate integrated multi-perspective evaluation of clustering algorithms using Pareto fronts

Givanna H. Putri[1,2,*], **Irena Koprinska**[1], **Thomas M. Ashhurst**[2,3], **Nicholas J.C. King**[2,3,4] **and Mark N. Read** [1,2,5,*]

[1]School of Computer Science, [2]Charles Perkins Centre, The University of Sydney, Sydney 2006, Australia, [3]Sydney Cytometry Facility, The University of Sydney and Centenary Institute, Sydney 2006, Australia[4]Discipline of Pathology and [5]Westmead Initiative, The University of Sydney, Sydney 2006, Australia

*To whom correspondence should be addressed.

Associate Editor: Anthony Mathelier

Received on September 11, 2020; revised on January 14, 2021; editorial decision on January 15, 2021; accepted on January 18, 2021

Abstract

Motivation: Many 'automated gating' algorithms now exist to cluster cytometry and single-cell sequencing data into discrete populations. Comparative algorithm evaluations on benchmark datasets rely either on a single performance metric, or a few metrics considered independently of one another. However, single metrics emphasize different aspects of clustering performance and do not rank clustering solutions in the same order. This underlies the lack of consensus between comparative studies regarding optimal clustering algorithms and undermines the translatability of results onto other non-benchmark datasets.

Results: We propose the Pareto fronts framework as an integrative evaluation protocol, wherein individual metrics are instead leveraged as complementary perspectives. Judged superior are algorithms that provide the best trade-off between the multiple metrics considered simultaneously. This yields a more comprehensive and complete view of clustering performance. Moreover, by broadly and systematically sampling algorithm parameter values using the Latin Hypercube sampling method, our evaluation protocol minimizes (un)fortunate parameter value selections as confounding factors. Furthermore, it reveals how meticulously each algorithm must be tuned in order to obtain good results, vital knowledge for users with novel data. We exemplify the protocol by conducting a comparative study between three clustering algorithms (ChronoClust, FlowSOM and Phenograph) using four common performance metrics applied across four cytometry benchmark datasets. To our knowledge, this is the first time Pareto fronts have been used to evaluate the performance of clustering algorithms in any application domain.

Availability and implementation: Implementation of our Pareto front methodology and all scripts and datasets to reproduce this article are available at https://github.com/ghar1821/ParetoBench.

Contact: givanna.haryonoputri@sydney.edu.au or mark.read@sydney.edu.au

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Cytometry and single-cell RNA sequencing (scRNAseq) technologies quantify myriad cellular features at single-cell resolutions, spanning expressed proteins to transcribed genes (Ashhurst *et al.*, 2017; Eberwine *et al.*, 1992; Hwang *et al.*, 2018). These technologies are revolutionizing biomedical research by revealing the body's inherently diverse and complex cellular systems (Spitzer and Nolan, 2016). Yet the increasingly high-dimensional nature of these technologies poses substantial analytical challenges.

Contemporary cytometry technologies can quantify up to 50 distinct cellular features simultaneously (Mair *et al.*, 2016). ScRNAseq can quantify many thousands. These technologies underpin ambitious initiatives such as the Human Cell Atlas (Regev *et al.*, 2017) which aims to map all cells in the human body, and has planned to sequence the entire RNA transcripts of 30–100 million cells (Hwang *et al.*, 2018). A critical analytical step is identifying discrete cellular populations amongst the cells analyzed. This has traditionally been performed through *manual gating*, in which polygons are manually drawn around perceived populations in a succession of two-dimensional density scatter plots. Each such plot represents the expression levels of two features for vast numbers of cells. The growing dimensionality of cytometry and scRNAseq technologies has rendered manual gating infeasible. Even a modest 20-feature analysis generates 190 distinct scatter plots. Manual gating suffers further drawbacks of user subjectivity and methodology, necessitating substantial user expertise, and thus poor reproducibility (Maecker *et al.*, 2012; Mair *et al.*, 2016; Saeys *et al.*, 2016). Small changes in gating can result in compounding and/or amplifying errors through downstream analytical stages, especially with small cell populations, potentially confounding correct biological interpretation.

These drawbacks have spurred the development of automated *clustering* approaches, from the field of unsupervised machine learning, to replace manual gating. This thriving research field has generated numerous solutions tailored towards cytometry and scRNAseq (Kiselev *et al.*, 2017; Levine *et al.*, 2015; Putri *et al.*, 2019; Samusik *et al.*, 2016; Satija *et al.*, 2015; Van Gassen *et al.*, 2015). The techniques differ in how they assemble clusters and the assumptions they make of the distribution of the data. Comparative studies are emerging that benchmark the available algorithms and offer advice to potential users. Proponents of new algorithms typically benchmark against the field, and tend to conclude their algorithm is superior (e.g. Levine *et al.*, 2015; Putri *et al.*, 2019; Samusik *et al.*, 2016; Van Gassen *et al.*, 2015). Independent comparative studies are also conducted, but again no consensus emerges (Aghaeepour *et al.*, 2016; Datta and Datta, 2003; Duò *et al.*, 2018; Freytag *et al.*, 2018; Soneson and Robinson, 2016; Thalamuthu *et al.*, 2006; Tian *et al.*, 2019; Weber and Robinson, 2016; Wiwie *et al.*, 2015; Yeung *et al.*, 2001). An algorithm that can accurately parse and analyze a dataset is critical to the unambiguous understanding of the biology captured. However, different approaches to comparing algorithms show none to perform best under all conditions, begging the question of how to choose the best algorithm for analysis.

We posit that the lack of consensus stems from methodological inconsistencies. First, while many metrics of clustering solution quality exist, comparison studies typically employ just one metric, or a small number analyzed independently of one another. Metrics differ in how they evaluate clustering, and thus which aspects of a clustering solution they are most sensitive to. These implicit biases could lead metrics to rank putative solutions differently, thus giving rise to the observed lack of consensus. Evaluation through a weighted sum of several metrics has been proposed (Weber *et al.*, 2019), but this lacks specificity as superior performance on one metric can mask poor performance on another. The technique is also sensitive to the weightings used, which are inevitably subjective and for which no guidance exists.

Second, a clustering algorithm's performance is highly sensitive to its parameter values, yet it is often unclear how these should be set for a given dataset. Comparison studies often contrast single 'optimal' representative algorithm results, selecting parameter values either manually, adopting those of prior applications, or through automated systematic exploration. Only the latter presents a convincing case that parameter values represent optimal performance. Importantly, these approaches do not reveal how readily high-quality performance is obtained. This is an important consideration for users seeking to analyze novel datasets where cell identities are not yet known. An algorithm which requires extensive parameter tuning may not be desirable. Indeed, it is not clear to what extent the dataset itself may influence the outcome of analysis by a particular algorithm.

A possible solution to all these challenges is the comprehensive evaluation of the proposed algorithms through several metrics of crucial performance characteristics, and the explicit capture of the trade-offs required to satisfy multiple potentially conflicting objectives in an optimal solution, wherein further improvement of any metric results in the decline of another. Such an approach has previously been employed in economics (Barr, 2020), logistics and engineering (Deb *et al.*, 2002) using Pareto fronts.

We present here a clustering performance evaluation framework that compares four prominent clustering metrics using this approach to overcome these challenges. This is exemplified through contrasting the popular FlowSOM (Van Gassen *et al.*, 2015) and Phenograph (Levine *et al.*, 2015) cytometry algorithms with the recent ChronoClust (Putri *et al.*, 2019). We show evidence that clustering metrics are discordant in their ranking of solutions. Importantly, the extent of disagreement varies by metric, dataset and algorithm, highlighting the necessity for comprehensive evaluations if results are to be generalizable. Our framework exploits these implicit metric biases, employing several metrics simultaneously to offer complementary perspectives on performance and judgement through 'wisdom of crowds'. This is accomplished through Pareto fronts, which make explicit the trade-offs that solutions present between the metrics. Algorithms judged superior generate solutions with a low degree of compromise: all metrics rank the solutions highly. Our framework employs systematic algorithm parameter sweeps. This (i) minimizes suboptimal parameter selection as a confounding factor and (ii) reveals how meticulously an algorithm's parameters must be tuned to generate optimal performance. We concisely illustrate the evaluation challenge and our solution through a study spanning three algorithms, four datasets and four clustering metrics.

Our framework enables a fundamental shift in how decisions are made of which algorithm to adopt: it resolves the ambiguities and inconsistencies that presently vex the field, and provides vital information to potential users on the ease of algorithm tuning.

2 Materials and methods

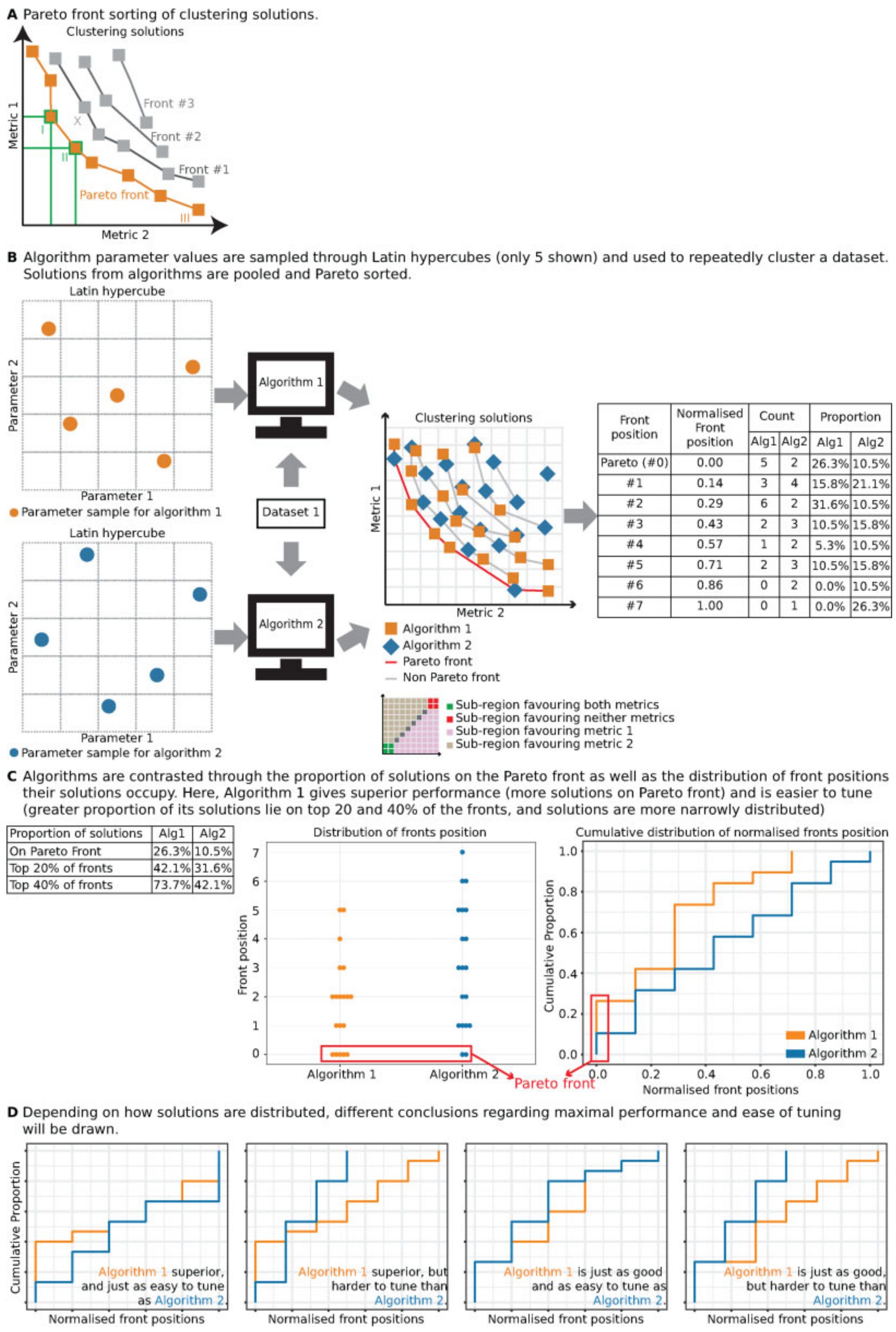
2.1 Complementary use of discordant clustering metrics through Pareto fronts

We propose a novel comparative evaluation framework for clustering algorithms which employs several metrics simultaneously to offer complementary perspectives on performance. This is accomplished through the use of Pareto fronts. Pareto fronts were first introduced by Pareto (1919), who used the concept to study economic efficiency: if a collection of policy options are differentially preferred amongst stakeholders, then Pareto fronts can identify those policies which represent the best compromises, i.e. those proposals for which everyone wins to the greatest extent possible. Pareto fronts have been adopted into multi-objective optimization techniques which seek optimal solutions to problems that have conflicting criteria (Read *et al.*, 2016a, b, 2020; Seada and Deb, 2016).

The Pareto front is the set of putative solutions amongst which an improvement in any one metric necessitates worsening in another (Read *et al.*, 2020), Fig. 1A. Conventionally in Pareto fronts, smaller metric values denote superior performance. The origin then represents a perfect (usually unattainable) solution. Maximization metrics can be converted through e.g. $1/x$. A solution will not reside on the Pareto front if another solution exists that bests it in all metrics. Solutions comprising the Pareto front are termed *Pareto-optimal*. A succession of fronts comprising *dominated* solutions can be established under this principle, which we conceptualize as offering *degrees of compromise*. Pareto-optimal solutions offer the smallest degree of compromise-given that some compromise is necessary, these solutions best satisfy all metrics' criteria. Subsequent fronts capture greater degrees of compromise; for each solution they contain, the preceding front captured at least one solution that *all* metrics have preferred.

Figure 1B illustrates the Pareto fronts concept as applied to hypothetical clustering solutions produced by two different algorithms and evaluated by two different performance metrics.

Clustering performance is highly sensitive to algorithm parameter values, and thus for comparative studies to be useful for users with new data, they should methodically explore parameter value selections as those users would have to. This is problem- and algorithm-specific. For some parameters, appropriate and generalizable default values may be known and can be adopted. For other parameters, exploration of potential values is warranted to find regions offering reasonable performance, as judged by the user. We suggest that conductors of comparative studies identify which parameters and ranges to explore in the same manner. The Pareto



A Pareto front sorting of clustering solutions.

B Algorithm parameter values are sampled through Latin hypercubes (only 5 shown) and used to repeatedly cluster a dataset. Solutions from algorithms are pooled and Pareto sorted.

Front position	Normalised Front position	Count		Proportion	
		Alg1	Alg2	Alg1	Alg2
Pareto (#0)	0.00	5	2	26.3%	10.5%
#1	0.14	3	4	15.8%	21.1%
#2	0.29	6	2	31.6%	10.5%
#3	0.43	2	3	10.5%	15.8%
#4	0.57	1	2	5.3%	10.5%
#5	0.71	2	3	10.5%	15.8%
#6	0.86	0	2	0.0%	10.5%
#7	1.00	0	1	0.0%	26.3%

C Algorithms are contrasted through the proportion of solutions on the Pareto front as well as the distribution of front positions their solutions occupy. Here, Algorithm 1 gives superior performance (more solutions on Pareto front) and is easier to tune (greater proportion of its solutions lie on top 20 and 40% of the fronts, and solutions are more narrowly distributed)

Proportion of solutions	Alg1	Alg2
On Pareto Front	26.3%	10.5%
Top 20% of fronts	42.1%	31.6%
Top 40% of fronts	73.7%	42.1%

D Depending on how solutions are distributed, different conclusions regarding maximal performance and ease of tuning will be drawn.

Cumulative Proportion
Algorithm 1 superior, and just as easy to tune as Algorithm 2.
Normalised front positions
Algorithm 1 superior, but harder to tune than Algorithm 2.
Normalised front positions
Algorithm 1 is just as good and as easy to tune as Algorithm 2.
Normalised front positions
Algorithm 1 is just as good, but harder to tune than Algorithm 2.
Normalised front positions

Fig. 1. A Pareto fronts framework for evaluating clustering performance by framing several metrics as conflicting assessment criteria, illustrated through a hypothetical example. (**A**) Clustering solutions are evaluated using several clustering metrics (only two shown). By convention in Pareto fronts, small metric values indicate superior performance. The Pareto front comprises *Pareto-optimal* solutions, those for which no other solution exists that improves on one metric without worsening on another. Solution II strikes a balanced trade-off between the two metrics, whereas solution III emphasizes performance on metric 1. Solution X is *dominated* by solutions I and II, both of which offer better improvement on all metrics. Solutions can be sorted into a succession of additional fronts by omitting the Pareto front and reapplying these same principles. (**B**) Parameter value samples are drawn for each algorithm, using Latin Hypercube sampling, and then used to generate clustering solutions for a given dataset (only two parameters shown). Pooling the solutions from competing algorithms and applying the Pareto front sort enables comparative analysis of algorithm performance. Algorithm 1 is judged superior as a greater proportion of its parameter space contributes to the Pareto front. The framework makes explicit how conflicting requirements of clustering performance are traded-off by algorithm solutions. Users can focus on given areas of the Pareto front if they favour given qualities of clustering performance or metrics over others. Normalizing front positions to the range [0, 1] allows the integration of Pareto sorting results from other datasets, as explained in the text. (**C**) Left: proportions of parameter space of algorithms 1 and 2 which lie on the Pareto front and the leading fronts (top 20% and 40%). Middle and Right: Swarm and Cumulative distribution plots portraying the distribution of solutions from each algorithm across the fronts. (**D**) The various interpretations that can be made of cumulative distribution plots. This analytical framework reveals: (1) which algorithm(s) are superior, by most readily contributing to the Pareto fronts, and (2) how meticulously algorithms must be tuned to obtain such performance, by revealing the distribution of solutions generated under algorithm parameter exploration across the fronts

Table 1. Summary of the 4 benchmark cytometry datasets used in our study. The original study authors assigned a high-confidence true label (cell type) to each cell in their respective datasets; some cells are not assigned labels, this is common in cytometry. All cells represent single, live cells. Percentages are of total cells in each dataset. Adapted from Weber and Robinson (2016).

	Levine_13dim	Levine_32dim	Samusik_01	Samusik_all
Total cell count	167,044	265,627	86,864	841,644
No. of markers	13	32	39	39
No. of populations	24	14	24	24
No. of assigned cells	81,747 (49%)	104,184 (49%)	53,173 (61%)	514,386 (64%)
Largest population size	13,964 (8.4%)	26,366 (9.9%)	13,607 (15.7%)	114,412 (13.6%)
Smallest population size	5 (0.003%)	304 (0.1%)	3 (0.003%)	29 (0.003%)

front framework determines the proportion of each algorithm's parameter space that contributes optimal (or near-optimal) performance. This indicates which algorithm offers outright superior performance, and also how readily good performance can be found.

To do this, the identified regions of parameter space must be adequately sampled. The number of samples taken must reflect the volume of parameter value space to be explored: large enough for sampling to be thorough but without generating duplicate samples. Broad sampling of algorithm parameter values can be accomplished via Latin Hypercube sampling (McKay, 1992). Latin Hypercube sampling segregates each parameter's range of values into discrete regions that are each sampled once only in a manner that minimizes correlations between parameters, Fig. 1B. The minimizing of correlations ensures that downstream analysis is sensitive to interactions between parameter values in dictating performance. The result is an efficient and extensive sampling of parameter space (Read *et al.*, 2020). By default, Latin Hypercube sampling employs uniform distributions when sampling parameters, but other distributions (e.g. Gaussian or exponential) can be employed to emphasize exploration of particular regions of parameter space (Marino *et al.*, 2008). Alternative sampling schemes, such as Grid Search, can also be employed.

The Pareto front framework gives a comprehensive analysis of performance in multi-faceted problems. If there is a reason to prefer a particular extent of trade-off, e.g. balanced performance on both metrics, or a preference for metric 1 over metric 2, solutions can be extracted from the respective regions of the Pareto front (Fig. 1B). Whilst we depict only two metrics for illustrative purposes, Pareto fronts can accommodate any number. Solutions arising from competing algorithms applied to a given dataset are pooled and sorted into Pareto fronts (Fig. 1B). An algorithm is superior if a greater proportion of its parameter space generates Pareto-optimal solutions (Fig. 1B and C). In this hypothetical example, Algorithm 1 is judged superior as 26.3% of its parameter space (5 out of 19 parameter samples) maps onto the Pareto front, as compared to 10.5% (2 out of 19) for Algorithm 2.

Our Pareto front framework reveals the ease with which algorithms are tuned to produce superior performance. Consider the distribution of each algorithm's solutions across the fronts, Fig. 1C and D. A greater proportion of Algorithm 1's parameter space maps onto the top 20% and 40% of the fronts (these thresholds can be changed to suit different needs). Likewise, Algorithm 1's solutions are more tightly distributed amongst the leading fronts, than those of Algorithm 2. Thus, the solutions obtained from Algorithm 1 tended to offer the lower degree of compromise. Whilst both Algorithms *can* produce Pareto-optimal solutions, the performance of Algorithm 2's solutions more readily deteriorate under sub-optimal parameter value selection. This is an important consideration for users choosing an algorithm to apply to their own unlabelled data; an algorithm that requires meticulous parameter tuning is undesirable if suitable parameters are unknown and cannot be easily ascertained. The greater proportion of Algorithm 1's solutions residing on the Pareto front indicates that it more readily produces superior performance under parameter tuning.

Analysis through our Pareto front framework can extend over several datasets, which can make for a more robust study (Weber and Robinson, 2016). As outlined in Figure 1B, our Pareto front framework procedure is applied to each dataset independently, pooling the performance of several algorithms. For each dataset, the Pareto front positions of the clustering solutions are normalized to lie in the range zero to one. The reason for this is that datasets can vary in the number of fronts that are generated, and normalization is needed to ensure that solutions (be they high or poor quality) on each dataset are judged likewise under pooling. This is followed by analysis of the clustering solutions, as shown in Fig. 1D.

2.2 Design of case study

To demonstrate the advantages of Pareto front-based analysis over monometric assessments, we employed three clustering algorithms: ChronoClust (Putri *et al.*, 2019), FlowSOM (Van Gassen *et al.*, 2015), and Phenograph (Levine *et al.*, 2015). They represent fundamentally different approaches to clustering. FlowSOM assembles a pre-specified number of clusters, whereas in ChronoClust the number of clusters emerges from the distribution and density of the data. Phenograph builds a graph connecting the data points and then partitions it into clusters.

We employed four benchmarking cytometry datasets previously used in a comparative study of clustering algorithms (Weber and Robinson, 2016), Table 1. Two datasets represent a mass cytometry investigation of progenitor-like cells in Acute Myeloid Leukaemia (*Levine_13dim* and *Levine_32dim* from Levine *et al.* (2015)). The remaining two datasets arose from a study of mouse bone marrow (*Samusik_01* and *Samusik_all* from Samusik *et al.* (2016)). These datasets were pre-processed using arc-sinh transformation with cofactor of 5 by Weber and Robinson (2016).

We employed four common supervised metrics of clustering quality: Accuracy, F1-score, V-measure (Rosenberg and Hirschberg, 2007) and Adjusted Rand Index (ARI) (Hubert and Arabie, 1985). These metrics are *supervised* in that they judge the clusters produced against some independently established 'ground truth'—a high-confidence labelling of which clusters (cellular populations) the data points should belong to, which we term *true labels*. The true labels were provided in the benchmarking datasets we adopted.

Accuracy and F1-score metrics require clusters (not just data points) to be assigned a true label. We employed the Hungarian method to provide these assignments, as has been performed in previous cytometry studies: Samusik *et al.* (2016) and Weber and Robinson (2016). Briefly, the Hungarian method determines a one-to-one assignment of labels to clusters so as to maximize the performance on the given metric, Accuracy or F1-score. We used $\beta = 0.6$ for V-measure throughout our study, favouring sensitivity for homogeneous clusters. A review of all four metrics can be found in Supplementary Section S1.

As is the convention with cytometry benchmarking studies (Weber and Robinson, 2016), we ran ChronoClust, FlowSOM and Phenograph on all cells in a dataset, but evaluate the quality of the clustering based on only cells assigned true labels. The scripts used to evaluate ChronoClust, FlowSOM and Phenograph clustering

Table 2. Number of unique parameter space samples produced by Latin Hypercube for each algorithm and dataset

Dataset	ChronoClust	FlowSOM	Phenograph
Levine_13dim	100	95	70
Levine_32dim	100	93	96
Samusik_01	100	99	70
Samusik_all	100	94	74

solutions are available from our GitHub repository https://github.com/ghar1821/ParetoBench.

Using Latin Hypercube sampling (McKay, 1992), 100 parameter value samples were generated for each of ChronoClust, FlowSOM and Phenograph for each dataset. Duplicate samplings of the same point in parameter space (owing to floats generated under Latin Hypercube resolving to the same value for integer-typed parameters) were discarded, Table 2.

For ChronoClust (Putri *et al.*, 2019), we varied the parameters $\delta, \epsilon, \upsilon$. For FlowSOM (Van Gassen *et al.*, 2015), we varied the number of meta clusters and the grid size. For Phenograph (Levine *et al.*, 2015), we varied the parameter k. These parameters are the most influential in determining the number and arrangement of clusters; we refer readers to the original publications of the algorithms for details regarding their parameters. Supplementary Material Section S2 details the range of values explored through Latin Hypercube sampling, and how they were chosen. We approached parameter selection as potential users of each algorithm would have to, accounting for advice by algorithm authors. The remaining parameter values which are not varied are supplied in Supplementary Tables S1, S3, and S5. Parameter space samplings used in this study are available at https://github.com/ghar1821/ParetoBench.

For each dataset independently, clustering solutions are sorted into Pareto fronts which adopt each supervised clustering performance metric as an orthogonal axis.

2.3 Statistical methods and implementations used

For clustering, we used the *Spectre*'s (Ashhurst *et al.*, 2020) FlowSOM implementation available from https://github.com/immunedynamics/spectre,

ChronoClust's Python implementation available from https://github.com/ghar1821/Chronoclust and Phenograph's Python implementation available from https://github.com/dpeerlab/PhenoGraph.

We used Latin Hypercube implementation from the *Spartan* R package (Alden *et al.*, 2013) to generate the parameter samples.

We used the following clustering performance metric implementations: *CoClust*'s implementation of Accuracy and its corresponding Hungarian cluster assignment (Role *et al.*, 2019), Weber's implementation of F1-score and its corresponding Hungarian cluster assignment method (Weber and Robinson, 2016), and the *scikit-learn* Python package implementations of ARI and V-measure (Pedregosa *et al.*, 2011). The implementation of Pareto front sorting was adopted from https://github.com/marknormanread/unsga3 (Read *et al.*, 2020).

Spearman rank correlation and Kolmogrov-Smirnov statistics were computed using *SciPy* Python package (Virtanen *et al.*, 2020). Permutational multivariate analysis of variance (PERMANOVA) (Anderson, 2014) was performed using the *adonis* function in the *vegan* R package (Oksanen *et al.*, 2019). PERMANOVA is a non-parametric variant of the ANOVA technique, and makes no assumptions concerning the distribution of the data.

3 Results

3.1 Optimal algorithm varies by metric and dataset

We hypothesized that different clustering metrics would favour different clustering algorithms. To test this, we applied ChronoClust, FlowSOM and Phenograph to the four cytometry datasets from Weber and Robinson (2016), summarized in Table 1. We drew 100 samples from the parameter spaces of each algorithm, removed duplicates and performed clustering with the remainder; this broad sampling of parameter space minimizes poor parameter value choices as a confounding factor. Figure 2A depicts the best performing clustering solutions as judged by F1-score for ChronoClust and FlowSOM, and highlights the divergent nature in which different algorithms partition cells into distinct cellular populations.

All three algorithms were capable of producing high-quality clusterings of the data. Figure 2B depicts the score of the best solutions as judged by each metric independently, for all algorithms as applied to each dataset. Given that each metric's highest score is 1, the results for Accuracy, ARI and V-measure were generally encouraging, with most solutions scoring around or over 0.8. Values obtained were lower for F1-score. Under V-measure, all three algorithms scored similarly. FlowSOM and Phenograph each generated the best score on two datasets. Algorithm scores diverged more substantially under Accuracy and ARI metrics, with Phenograph producing notably poor solutions on the *Levine_32dim* dataset and yet the best performances on the *Levine_13dim* and *Samusik_all* datasets. FlowSOM generated the best solutions for the remaining two datasets under both metrics. Under F1-score, ChronoClust generated the best solutions for two datasets and the worst solutions for the other two datasets.

It is difficult to converge on an optimal algorithm with these data. FlowSOM was consistently competitive, but not always optimal. Conclusions drawn from these data would depend on which metric was used and which dataset was being examined, and if taking consideration of all of them, would be somewhat ambivalent.

3.2 Imperfect concordance between supervised clustering quality metrics

We next investigated whether different clustering metrics agreed on which individual solution was optimal for a given dataset. We cross-referenced the rankings of solutions produced under each metric. We found for all algorithms that the solutions judged best by F1-score are rarely so when judged by other metrics, as shown in the rankings at the bottom of the bars in Figure 2B and Supplementary Table S7. This was not specific to the F1-score. We found similar discord when ranking solutions by each of the other metrics, Supplementary Tables S8, S9 and S10. The discordance was more pronounced for Phenograph and FlowSOM, where optimal solutions by one metric were ranked as low as 87th and 67th respectively by others, than for ChronoClust (worst rank of 19th). These results show little agreement between metrics as to which clustering solutions are optimal, and highlight how the degree of discord varies by algorithm.

To explore these observations further, we pooled all clustering solutions' metric scores from all algorithms and datasets, and then calculated the Spearman's (rank) correlation coefficient (Rs) between pairs of metrics. As shown in Figure 2C, we found that in general there exists a positive, though imperfect, correlation for all supervised metrics. Agreement was strongest between Accuracy and ARI ($Rs = 0.97$) and weakest for Accuracy and F1-score ($Rs = 0.71$). Thus, although we found metrics to disagree over which clustering solutions were strictly optimal, they did agree in broader terms over which solutions were better or worse.

We were curious as to whether the degree of concordance between metrics held generally, as suggested in Figure 2C, or whether this varied by dataset and/or clustering algorithm. We calculated Spearman's correlations (Rs) between metric scores of clustering solutions for each algorithm and dataset independently. We used the permutational multivariate analysis of variance (PERMANOVA) to analyze whether Rs correlations varied by (1) dataset, (2) algorithm, (3) combination of metrics, and all pairwise interactions between these three factors, Fig. 3A. We found statistically significant differences by metric-pairings, dataset, algorithm and their interactions, Fig. 3B. We parsed Rs values across these experimental factors, Fig. 3C–G. The most congruent pairs of metrics were Accuracy and ARI, Accuracy and V-measure, and ARI and V-measure, Fig. 3C.

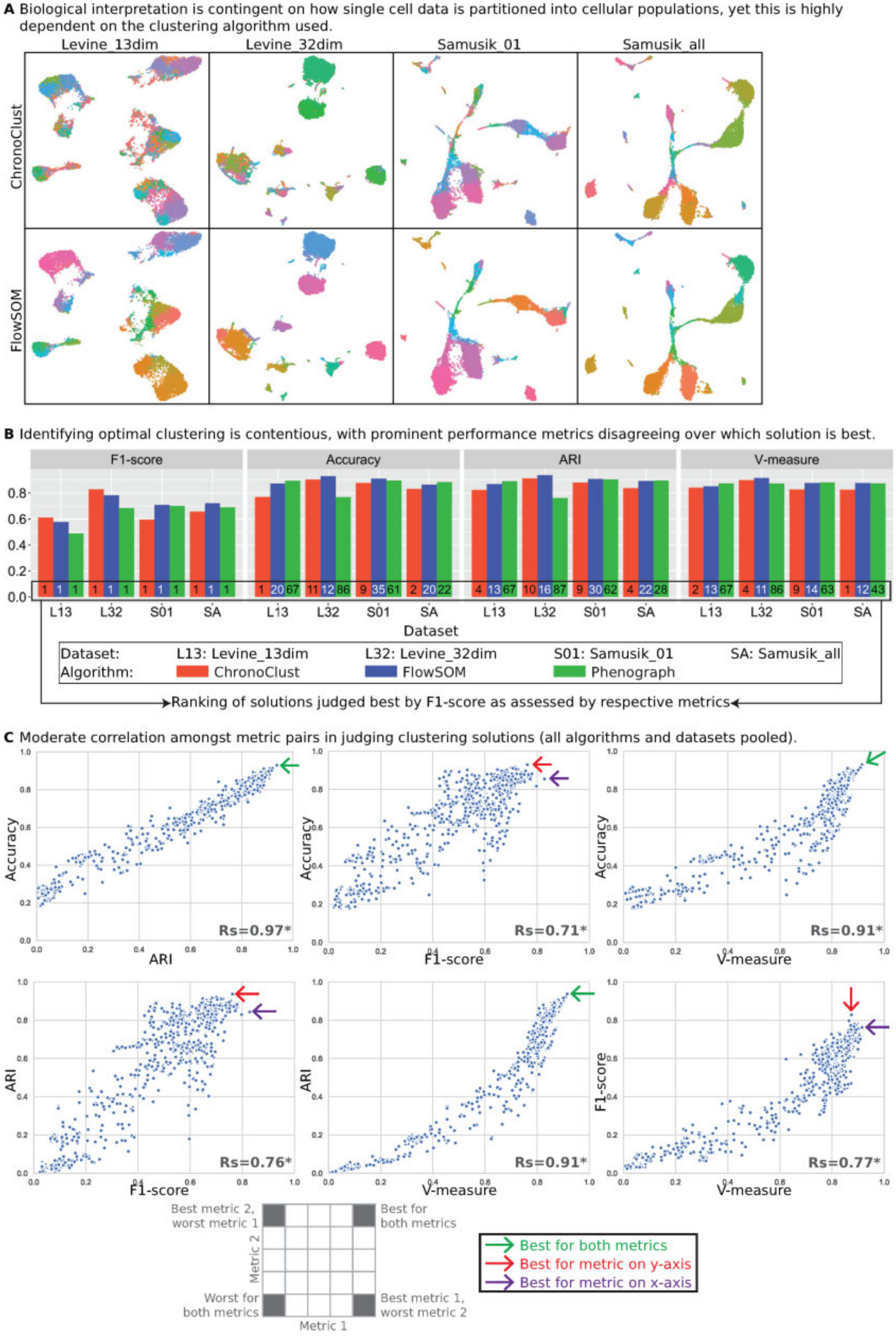
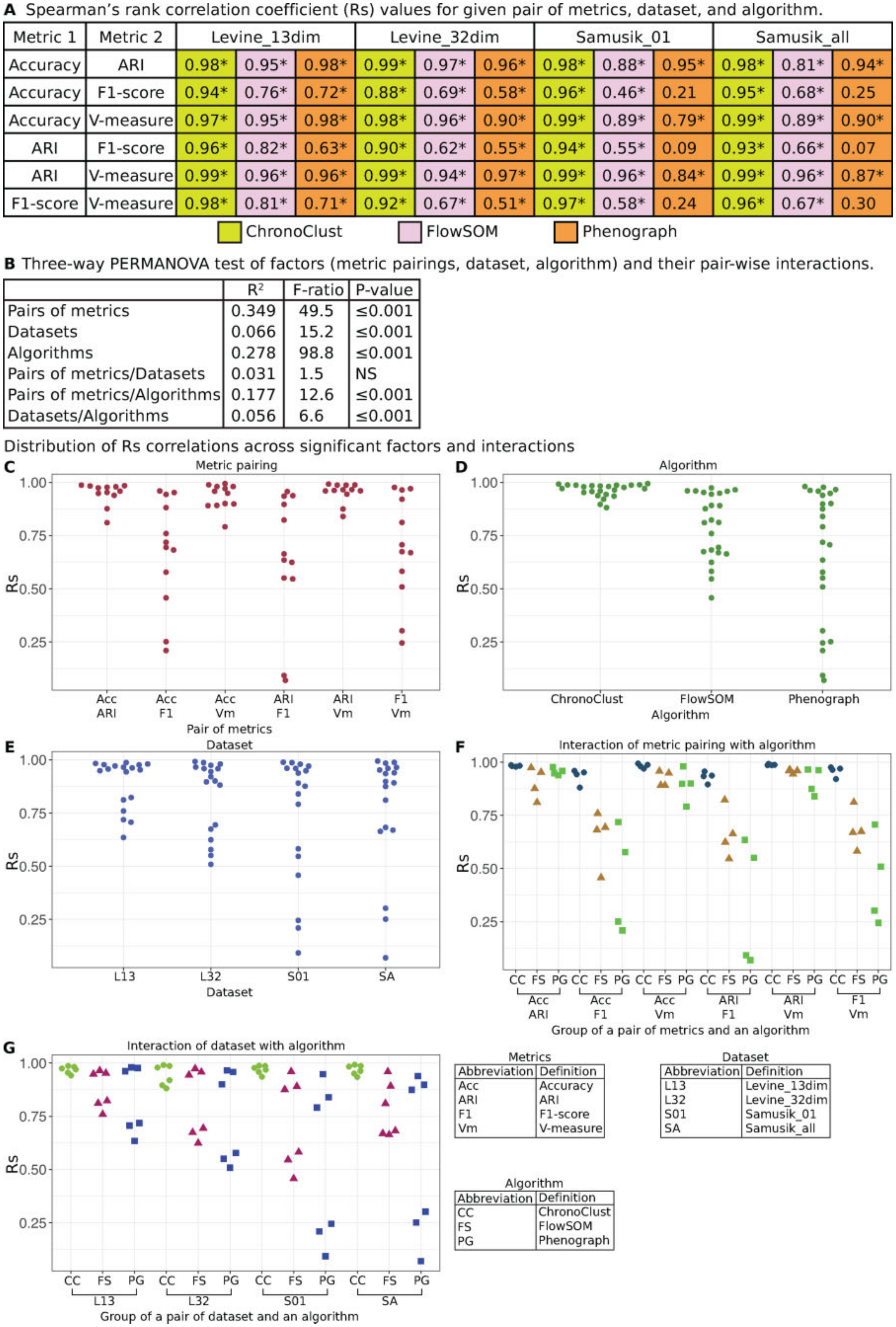


Fig. 2. Appropriate choice of clustering algorithm for assembling single-cell quantifications into cellular populations is critical for unambiguous biological interpretation, yet this choice is contentious as clustering quality metrics pass discordant judgements. (**A**) UMAP plots (McInnes *et al.*, 2018) depicting the structure of the datasets used in our case study (Table 1, data subsampled to 10,000 cells). Within each plot, colours denote cellular cluster memberships. Clustering solutions shown represent those with the best F1-scores from up to 100 distinct parameter value samplings for each algorithm. Within each dataset, the partitioning of cells into clusters representing cellular populations varies by algorithm. The nature of this clustering can profoundly impact downstream analysis outcomes; biological interpretations are thus dependent on the choice of clustering algorithm. (**B**) Bar charts depicting the highest scoring clustering solutions as judged independently by four prominent supervised clustering quality metrics, from up to 100 distinct clustering algorithm parameter value combinations per algorithm, per dataset. For all metrics, scores of 1.0 represent optimal clustering. Values at the bottom of each bar represent the rank of the solutions favoured under F1-score when judged by the indicated metric; the top-ranked solution by F1-score, on a given dataset, is not top-ranked by other metrics. It is thus unclear which algorithm, and which parameter values, one should adopt. (**C**) Scatter plots illustrating the Spearman's rank correlation coefficient (*Rs*) between the scores given by pairs of supervised metrics. The data represent metric scores given to clustering results solutions from 1089 unique parameter value sets: up to 100 for each of four datasets, for each algorithm. Discordance in metric judgements is thus not confined to the best performing clustering solutions, but extends throughout the range of performances exhibited



A Spearman's rank correlation coefficient (Rs) values for given pair of metrics, dataset, and algorithm.

Metric 1	Metric 2	Levine_13dim			Levine_32dim			Samusik_01			Samusik_all		
Accuracy	ARI	0.98*	0.95*	0.98*	0.99*	0.97*	0.96*	0.98*	0.88*	0.95*	0.98*	0.81*	0.94*
Accuracy	F1-score	0.94*	0.76*	0.72*	0.88*	0.69*	0.58*	0.96*	0.46*	0.21	0.95*	0.68*	0.25
Accuracy	V-measure	0.97*	0.95*	0.98*	0.98*	0.96*	0.90*	0.99*	0.89*	0.79*	0.99*	0.89*	0.90*
ARI	F1-score	0.96*	0.82*	0.63*	0.90*	0.62*	0.55*	0.94*	0.55*	0.09	0.93*	0.66*	0.07
ARI	V-measure	0.99*	0.96*	0.96*	0.99*	0.94*	0.97*	0.99*	0.96*	0.84*	0.99*	0.96*	0.87*
F1-score	V-measure	0.98*	0.81*	0.71*	0.92*	0.67*	0.51*	0.97*	0.58*	0.24	0.96*	0.67*	0.30

🟩 ChronoClust ▢ FlowSOM 🟧 Phenograph

B Three-way PERMANOVA test of factors (metric pairings, dataset, algorithm) and their pair-wise interactions.

	R^2	F-ratio	P-value
Pairs of metrics	0.349	49.5	≤0.001
Datasets	0.066	15.2	≤0.001
Algorithms	0.278	98.8	≤0.001
Pairs of metrics/Datasets	0.031	1.5	NS
Pairs of metrics/Algorithms	0.177	12.6	≤0.001
Datasets/Algorithms	0.056	6.6	≤0.001

Distribution of Rs correlations across significant factors and interactions

Metrics

Abbreviation	Definition
Acc	Accuracy
ARI	ARI
F1	F1-score
Vm	V-measure

Dataset

Abbreviation	Definition
L13	Levine_13dim
L32	Levine_32dim
S01	Samusik_01
SA	Samusik_all

Algorithm

Abbreviation	Definition
CC	ChronoClust
FS	FlowSOM
PG	Phenograph

Fig. 3. (**A**) Spearman's correlations (*Rs*) between pairs of metrics for ChronoClust's (green), FlowSOM's (pink), and Phenograph's (orange) clustering solutions on each dataset. * indicates *P*-value < 0.005 for correlations. (**B**) Statistical values under permutational analysis of variance (PERMANOVA) to parse effects amongst experimental variables and their pairwise interactions. There was insufficient data to support an analysis of interactions amongst all three experimental variables. (**C–G**) Swarm plots illustrating the distribution of *Rs* values across different experimental variables found to be statistically significant in B

All other pairings exhibited more modest agreement. This clearly interacted with the algorithm being analyzed: metrics were far more congruent when assessing ChronoClust than FlowSOM or Phenograph, Fig. 3D and F. Additionally, we also found statistically significant effect by dataset, Fig. 3E, and by interaction of dataset with algorithm: the extent of disagreement between metrics was sensitive to which algorithm was being analyzed on which given dataset, Fig. 3G.

In summary, our data advise caution when conducting and interpreting comparative clustering studies. First, different metrics present biases in how they judge clustering solutions: results are a function of metrics employed, not only the performances of the algorithms studied. Basing comparative studies on single metrics alone is not advised unless there is firm reason to believe the chosen metric's biases align well with desired qualities of the clustering solution, and that other qualities are not relevant. Further, not only do metrics exhibit disagreement in which clustering solutions are best, the extent of disagreement is particular to the algorithm, and which dataset the algorithm was run on. Thus, for those examining comparative studies with a view to choosing an approach for their own study, results may not prove representative. The choice of metric, and dataset(s), used can immediately bias the analysis towards a particular algorithm.

3.3 Multi-perspective evaluation through Pareto fronts framework

We developed our Pareto front evaluation framework as a mean to integrate the complementary perspectives that different metrics and datasets offer over algorithm performance (Fig. 1, Section 2.1). No single solution, from any algorithm, was simultaneously top-ranked by all four metrics (Supplementary Material Tables S7–S10). Pareto fronts interpret such discords into trade-offs and degrees of compromise with respect to several criteria. The leading Pareto front (front #0) captures solutions necessitating minimal compromise: no other solutions were found that all metrics simultaneously prefer, i.e. these solutions best satisfy all metrics' criteria simultaneously. The succession of fronts that follow (front #1, etc.) capture progressively greater degrees of compromise, wherein a solution favoured by one metric is more strongly disfavoured by others. Our framework maps sampled algorithm parameter space onto these fronts. Thus, we ascertain which algorithms are capable of producing optimal performance (front #0), and also how readily high-quality performance can be generated (greater proportion of parameter space maps onto leading fronts). The latter reveals how meticulously an algorithm must be tuned, and here we considered the leading 10% and 33% of fronts to focus on particularly high-quality solutions. Desirable are algorithms that generate high-quality solutions from large portions of their explored parameter space.

We applied our Pareto front framework (Fig. 1) to the four clustering metric scores of solutions generated by Chronoclust, FlowSOM and Phenograph. Pareto front sorting was performed on each dataset independently, pooling solutions from all three algorithms, Table 3 and Fig. 4. *All datasets* represents the pooling of an algorithm's results from each dataset into a single distribution.

From Table 3, it is immediately evident that FlowSOM and Phenograph outperform ChronoClust in generating optimal solutions. A far greater proportion of parameter space samplings resides on the Pareto front for FlowSOM (∼5%) and Phenograph (∼4%) than for ChronoClust (< 1%). ChronoClust did not contribute any solutions to the leading Pareto front (front #0) for either of the *Samusik* datasets. ChronoClust was also the hardest algorithm to tune. For example, consider the pooled results, *all datasets*—only 3% and 20% of ChronoClust solutions mapped onto the top 10% and 33% of fronts, respectively. This contrasts with 43% and 74% for FlowSOM, and 34% and 85% for Phenograph. As shown in Figure 4, ChronoClust's parameter space mapped onto a much wider range of fronts, and was by far the most concentrated towards the trailing half of the fronts (∼60% of solutions in the bottom 50% of fronts).

Table 3. The proportion of parameter space explored, for each of ChronoClust, FlowSOM, and Phenograph, that maps onto the optimal Pareto front and the top 10% and 33% of fronts. The best performing algorithms are indicated in bold font.

Algorithm	Proportion of parameter space that maps onto		
	Pareto front (%)	Top 10% of fronts (%)	Top 33% of fronts (%)
All datasets			
ChronoClust	0.8	3.3	20.3
FlowSOM	**5.0**	**43.3**	73.5
Phenograph	4.2	34.1	**84.7**
Levine_13dim			
ChronoClust	2.0	4.0	11.0
FlowSOM	6.3	26.3	69.5
Phenograph	**7.1**	**30.0**	**88.6**
Levine_32dim			
ChronoClust	1.0	7.0	37.0
FlowSOM	**2.2**	**34.4**	44.1
Phenograph	0.0	0.0	**56.8**
Samusik_01			
ChronoClust	0.0	0.0	9.0
FlowSOM	**5.1**	**40.4**	83.8
Phenograph	4.3	34.3	**94.3**
Samusik_all			
ChronoClust	0.0	0.0	17.0
FlowSOM	6.4	46.8	84.0
Phenograph	**6.8**	**47.9**	**97.3**

FlowSOM and Phenograph offered quite comparable performances for three of the datasets, *Levine_13dim* and both *Samusik* datasets. Their tendencies to contribute to the leading front (front #0) was similar with at most 0.8% difference. With respect to ease of tuning, Phenograph consistently contributed more solutions to the top 33%, but neither was unanimously dominant for the top 10%. Holistically, the mappings of their respective parameter spaces across fronts were statistically indistinguishable for these three datasets (Fig. 4).

For the remaining *Levine_32* dataset, we found a striking divergence in FlowSOM and Phenograph performances. Here, Phenograph failed to map at all onto the leading 15% of fronts, whereas 40% of FlowSOM solutions fell within this range (Fig. 4).

In conclusion, under our evaluation framework, ChronoClust was evidently the worst performing algorithm and the hardest to tune. FlowSOM and Phenograph offered comparable, leading performance for three datasets. But for the remaining dataset, Phenograph performance was strikingly substandard to that of FlowSOM, and even ChronoClust contributed better quality solutions. In our view, based on these data, FlowSOM is the superior algorithm.

4 Discussion

Comparative studies of clustering algorithms rely on supervised clustering performance metrics to judge clustering quality. Such studies often rely on the performance judgement(s) of either a single metric (Aghaeepour *et al.*, 2016; Weber and Robinson, 2016; Yeung *et al.*, 2001) or several metrics applied independently and not analyzed in an integrated fashion (Datta and Datta, 2003; Duò *et al.*, 2018; Freytag *et al.*, 2018; Soneson and Robinson, 2016; Thalamuthu *et al.*, 2006; Tian *et al.*, 2019; Wiwie *et al.*, 2015). However, metrics are not impartial, they each emphasize different aspects of clustering performance, and as such do not rank solutions in the same order. No single solution was judged unanimously optimal by the four metrics we explored. Further, this disagreement is particular to the algorithm and the dataset used. These discrepancies are particularly problematic in the case of exploratory datasets, where true cluster

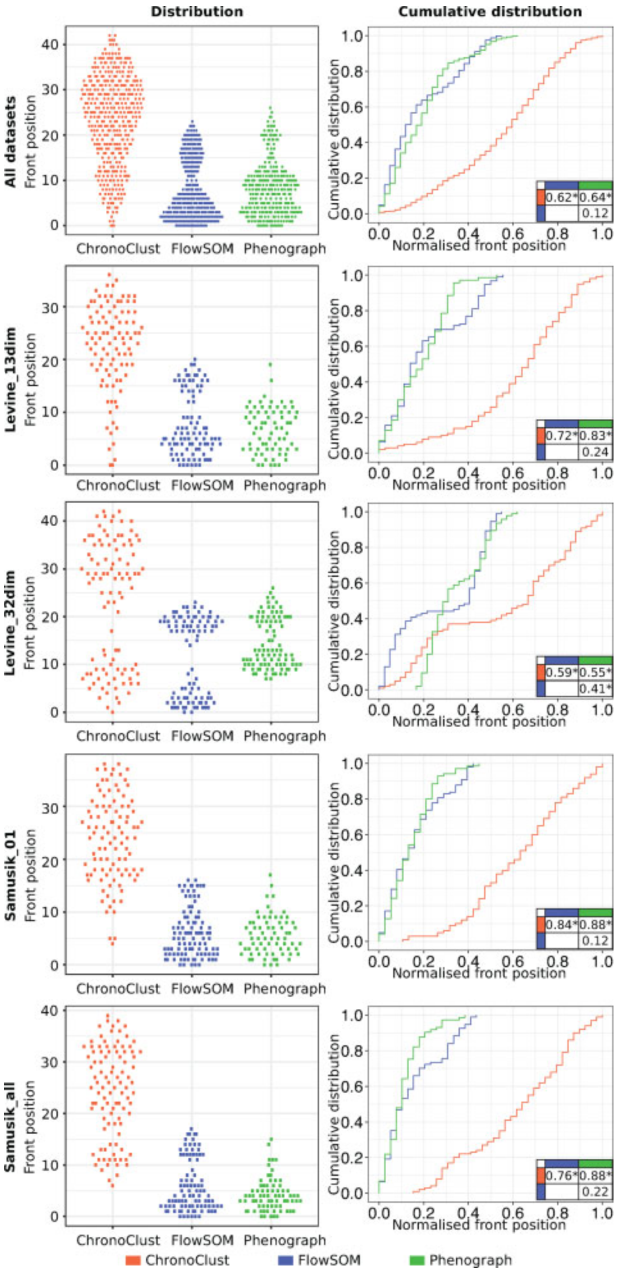


Fig. 4. Left: distribution of front positions of ChronoClust's, FlowSOM's, and Phenograph's clustering solutions. Right: corresponding cumulative distributions of normalized front positions. The Pareto fronts framework is applied to each dataset independently, and normalized positions are then pooled to create the 'all datasets' graph (top). Kolmogorov–Smirnov (KS) statistics are reported as a measure of effect magnitude. * indicates P-value < 0.005. Summary of comparison is available in Table 3

identities are incompletely established and may undergo differential changes over time, and for which analysis cannot therefore simply be related to known true cluster identities. As such, the conclusions drawn vary by clustering metric, and the discordance between them would weaken the conviction of the overall conclusions.

We proposed Pareto fronts as a framework for integrating the disparate perspectives offered by different metrics over different datasets, turning their implicit biases into a strength through a complementary use and an elegantly unified analysis. We demonstrated this transformative, novel approach for comparative algorithm analysis through a study of three algorithms across four datasets using four metrics. The Pareto front approach gives a more robust and comprehensive assessment of clustering performance and is analogous to the concepts of 'wisdom of crowds' or 'trial by jury'. This same principle underpins the use of multiple models as ensembles in supervised machine learning. Under our Pareto front methodology, FlowSOM emerged unequivocally as the superior algorithm, whereas evaluations based on single metrics alone prompted only a marginal and non-unanimous preference for FlowSOM. Why the discrepancy? Contrasting algorithms on the basis of best solutions independently of one another, as highlighted by single metrics, is akin to contrasting solutions on the extremes of the Pareto front only. We contend that optimal balancing of multiple criteria (metrics) of clustering quality *simultaneously* makes for a far more compelling assessment of algorithm performance. The framework makes explicit the trade-offs, or 'degree of compromise', that algorithms' solutions exhibit across metrics. Superior algorithms necessitate little compromise: all metrics agree that the solutions produced are of the highest quality. This is what our Pareto front framework enables, and the conclusions it generated were far more clear-cut. Clustering solutions under each algorithm arose from a systematic sampling of algorithm parameter space. This (1) minimizes (un)fortunate parameter value selections as a confounding factor in the comparisons and (2) allows assessment of how meticulous parameter values need to be tuned to obtain the best performance, an important consideration for potential users. To our knowledge, this is the first time Pareto fronts have been used to evaluate the performance of clustering algorithms in any application domain.

An important consideration in all comparative study methodologies is the selection of regions of algorithm parameter value space to explore. Algorithm performances are sensitive to parameter values, and poor choices will lead to under-performance which will impact comparative study results. In our Pareto front methodology, distributing samples of parameter value space into regions of poor algorithm performance can lessen the opportunity for that algorithm to contribute to the Pareto front, and render it seemingly difficult to tune. The onus is on the conductor of a comparative study to make an honest and transparent effort to find and explore appropriate regions of parameter space. Comparative studies aim to supply potential users, who wish to cluster their data, with helpful advice. As such, we advocate that comparative studies approach the algorithms in the same way that a user with novel data would, thus rendering findings most relevant to the user's situation. Potential users would make use of domain knowledge (e.g. how many cell populations they seek to uncover and to what extent they are prepared to over-/under-cluster), and advice by the algorithm's authors or other prior studies concerning appropriate or default parameter value settings. It is also a standard practice to engage in some preliminary parameter space exploration to identify seemingly promising regions for their dataset. This was the approach we adopted, tailoring the ranges of parameters explored to each dataset in an honest effort to seek quality performance.

Some metrics, Accuracy and F1-score here, require clusters to be associated with a true label. Here, we used the Hungarian assignment method, as employed in previous landmark cytometry studies (Samusik *et al.*, 2016; Weber and Robinson, 2016). We note that there are alternative methods for providing these assignments, e.g. assigning a cluster: (1) the majority label of its captured datapoints; (2) the label of the nearest ground truth cluster, by centroids as found in Putri *et al.* (2019) and (3) the label which maximizes a metric score (Aghaeepour *et al.*, 2013; Steinbach *et al.*, 2000). Metric scores may be sensitive to this choice of assignment method; however, the choice can be ruled out as a confounding factor if each alternative is incorporated into the Pareto front framework.

The context of clustering algorithms in cytometry and single-cell sequencing is to automate the process of manual gating, and comparative studies serve to advise the field on which algorithm to use. However, our findings show that metrics' judgements are sensitive to the specific algorithm and the dataset it was run on. This suggests that a given algorithm's superior performance on a given benchmark dataset may not extend onto a user's novel dataset. This underscores the need for comparative studies to employ multiple datasets and metrics. Ideally, however, we would develop the means to identify datasets with similar qualities to one another, independent of clustering, and thus users could select algorithms based on performance on the most similar benchmarking dataset(s) to their own. To our knowledge, this capacity does not currently exist.

Lastly, clustering algorithms are sensitive to parameter values, and widely applicable 'default' values that will give desirable performance are often unknown. Users cannot rely on supervised clustering metrics to guide parameter value choices if those metrics require labelled data arising from the very process clustering seeks to automate. We see tremendous value in identifying *unsupervised* metrics, requiring no label, that indicate when clustering performance approximates that which an expert would generate manually. To date, no such metric has been found. Wiwie *et al.* (2015) did evaluate several clustering tools using three unsupervised metrics and found only moderate correlation between the unsupervised metric *silhouette score* (Rousseeuw, 1987) and the supervised metric F1-score. We hypothesize that the use of several unsupervised metrics through the Pareto front framework would prove more fruitful.

Funding

G.H.P. was supported by the Australian Government Research Training Program (RTP) Scholarship. T.M.A. was supported by the International Society for Advancement of Cytometry Marylou Ingram Scholars.

Conflict of Interest: none declared.

References

Aghaeepour,N. *et al.* (2013) Critical assessment of automated flow cytometry data analysis techniques. *Nat. Methods*, **10**, 228–238.

Aghaeepour,N. *et al.* (2016) A benchmark for evaluation of algorithms for identification of cellular correlates of clinical outcomes. *Cytometry Part A*, **89**, 16–21.

Alden,K. *et al.* (2013) Spartan: a comprehensive tool for understanding uncertainty in simulations of biological systems. *PLoS Comput. Biol.*, **9**, e1002916.

Anderson,M.J. (2014) Permutational multivariate analysis of variance (PERMANOVA). In: Balakrishnan, N. *et al.* (eds.), *Wiley StatsRef: Statistics Reference Online*. Wiley, Hoboken, NJ, pp. 1–15.

Ashhurst,T.M. *et al.* (2017) High-dimensional fluorescence cytometry. *Curr. Protoc. Immunol.*, **119**, 5–8.

Ashhurst,T.M. *et al.* (2020) Integration, exploration, and analysis of high-dimensional single-cell cytometry data using Spectre. https://doi.org/10.1101/2020.10.22.349563

Barr,N. (2020) *Economics of the Welfare State*. Oxford University Press, USA.

Datta,S. and Datta,S. (2003) Comparisons and validation of statistical clustering techniques for microarray gene expression data. *Bioinformatics*, **19**, 459–466.

Deb,K. *et al.* (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.*, **6**, 182–197.

Duò,A. *et al.* (2018) A systematic performance evaluation of clustering methods for single-cell RNA-seq data. *F1000Res.*, **7**, 1141.

Eberwine,J. *et al.* (1992) Analysis of gene expression in single live neurons. *Proc. Natl. Acad. Sci. U. S. A.*, **89**, 3010–3014.

Freytag,S. *et al.* (2018) Comparison of clustering tools in r for medium-sized 10x genomics single-cell RNA-sequencing data. *F1000Res.*, **7**, 1297.

Hubert,L. and Arabie,P. (1985) Comparing partitions. *J. Class.*, **2**, 193–218.

Hwang,B. *et al.* (2018) Single-cell RNA sequencing technologies and bioinformatics pipelines. *Exp. Mol. Med.*, **50**, 1–14.

Steinbach,M. *et al.* (2000) A comparison of document clustering techniques. In Proceedings of the International KDD Workshop on Text Mining.

Kiselev,V.Y. *et al.* (2017) SC3: consensus clustering of single-cell RNA-seq data. *Nat. Methods*, **14**, 483–486.

Levine,J.H. *et al.* (2015) Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis. *Cell*, **162**, 184–197.

Maecker,H.T. *et al.* (2012) Standardizing immunophenotyping for the human immunology project. *Nat. Rev. Immunol.*, **12**, 191–200.

Mair,F. *et al.* (2016) The end of gating? An introduction to automated analysis of high dimensional cytometry data. *Eur. J. Immunol.*, **46**, 34–43.

Marino,S. *et al.* (2008) A methodology for performing global uncertainty and sensitivity analysis in systems biology. *J. Theor. Biol.*, **254**, 178–196.

McInnes,L. *et al.* (2018) UMAP: uniform manifold approximation and projection for dimension reduction. *J. Stat. Softw,* **3**, 861.

McKay,M.D. (1992) Latin hypercube sampling as a tool in uncertainty analysis of computer models. *Technical report.* Los Alamos National Lab., NM, USA.

Oksanen,J. *et al.* (2019) vegan: community Ecology Package. R package version 2.5-6.

Pareto,V. (1919) *Manuale di Economia Politica Con Una Introduzione Alla Scienza Sociale*, Vol. **13**. Società editrice libraria.

Pedregosa,F. *et al.* (2011) Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.*, **12**, 2825–2830.

Putri,G.H. *et al.* (2019) Chronoclust: density-based clustering and cluster tracking in high-dimensional time-series data. *Knowl. Based Syst.*, **174**, 9–26.

Read,M.N. *et al.* (2016a) Automated multi-objective calibration of biological agent-based simulations. *J. R. Soc. Interface*, **13**, 20160543.

Read,M.N. *et al.* (2016b) Leukocyte motility models assessed through simulation and multi-objective optimization-based model selection. *PLoS Comput. Biol.*, **12**, e1005082.

Read,M.N. *et al.* (2020) Strategies for calibrating models of biology. *Brief. Bioinform.*, **21**, 24–35.

Regev,A. *et al.* (2017) Science forum: the human cell atlas. *elife*, **6**, e27041.

Role,F. *et al.* (2019) CoClust: A Python Package for Co-Clustering. *J. Stat. Softw.*, **88**, 1–29.

Rosenberg,A. and Hirschberg,J. (2007) V-measure: a conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Association for Computational Linguistics, Prague, Czech Republic, pp. 410–420.

Rousseeuw,P.J. (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math*., **20**, 53–65.

Saeys,Y. *et al.* (2016) Computational flow cytometry: helping to make sense of high-dimensional immunology data. *Nat. Rev. Immunol.*, **16**, 449–462.

Samusik,N. *et al.* (2016) Automated mapping of phenotype space with single-cell data. *Nat. Methods*, **13**, 493–496.

Satija,R. *et al.* (2015) Spatial reconstruction of single-cell gene expression data. *Nat. Biotechnol.*, **33**, 495–502.

Seada,H. and Deb,K. (2016) A unified evolutionary optimization procedure for single, multiple, and many objectives. *IEEE Trans. Evol. Comput.*, **20**, 358–369.

Soneson,C. and Robinson,M.D. (2016) iCOBRA: open, reproducible, standardized and live method benchmarking. *Nat. Methods*, **13**, 283–283.

Spitzer,M.H. and Nolan,G.P. (2016) Mass cytometry: single cells, many features. *Cell*, **165**, 780–791.

Thalamuthu,A. *et al.* (2006) Evaluation and comparison of gene clustering methods in microarray analysis. *Bioinformatics*, **22**, 2405–2412.

Tian,L. *et al.* (2019) Benchmarking single cell RNA-sequencing analysis pipelines using mixture control experiments. *Nat. Methods*, **16**, 479–487.

Van Gassen,S. *et al.* (2015) FlowSOM: using self-organizing maps for visualization and interpretation of cytometry data: flowSOM. *Cytometry Part A*, **87**, 636–645.

Virtanen,P. *et al.* (2020) SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods*, **17**, 261–272.

Weber,L.M. and Robinson,M.D. (2016) Comparison of clustering methods for high-dimensional single-cell flow and mass cytometry data: comparison of High-Dim. cytometry clustering methods. *Cytometry Part A*, **89**, 1084–1096.

Weber,L.M. *et al.* (2019) Essential guidelines for computational method benchmarking. *Genome Biol.*, **20**, 125.

Wiwie,C. *et al.* (2015) Comparing the performance of biomedical clustering methods. *Nat. Methods*, **12**, 1033–1038.

Yeung,K.Y. *et al.* (2001) Validating clustering for gene expression data. *Bioinformatics*, **17**, 309–318.