Defining confidence in flow cytometry automated data analysis software platforms

Defining Confidence in Flow Cytometry Automated Data Analysis Software Platforms

by

Melissa Cheung

A Doctoral Thesis

Submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy of Loughborough University

July 2022

Abstract

The development of flow cytometry data analysis computational tools in recent years has the potential to reduce the variation arising from manual gating and improve the quality of cell characterisations performed within academic, clinical and biomanufacturing settings. However, there is a need to understand the uncertainty of measurements from automated tools, alongside a need for benchmarking datasets with ground truth that enable systematic comparisons to be made between these tools. This thesis investigates the cell identification outputs of software that utilise different classes of clustering algorithms, with a focus on implementing highly controlled synthetic datasets for performance evaluation.

A literature survey was conducted to identify the most cited tools, enabling the selection of the most relevant ones representative of different unsupervised clustering techniques: Flock2, flowMeans, FlowSOM, PhenoGraph, SPADE1, SPADE3 and SWIFT. Synthetic flow cytometry datasets were designed and generated with specific data characteristics of separation, normal/skew distributions, rarity and noise elements, and demonstrated to be credible substitutes for real cell data. These synthetic datasets were applied to the different software tools to determine the accuracy and repeatability of absolute cell counts. The results demonstrated how outputs from software analysing the same reference synthetic dataset vary considerably with accuracy deteriorating as the clusters overlapped and the separation index fell below zero. Moreover, SWIFT was found to be more negatively affected than other software in the presence of skewed cell populations. Assessment of rare cell detection revealed most software failed to consistently achieve a limit of detection of 100 cells in 10^6 events (0.01%). The addition of noise events resulted in a decrease in performance from all software, most significantly for FlowSOM. Furthermore, an automated versus manual comparison study carried out using a CD34+ stem cell dataset revealed higher variability from automated outputs compared to manual ones (mean coefficient of variations of 96% and 54%, respectively), and a weak correlation between the two methods (r=0.33) when analysing less well-separated cell populations.

This work has illustrated how the generation of novel synthetic flow cytometry datasets, and their application in comparison studies, has allowed the performance limitations of different automated software tools to be uncovered. The synthetic datasets benefit from having known ground truth not obtainable from real world datasets, therefore have potential utility as digital reference materials, possibly leading to enhanced measurement confidence in automated cell characterisations and enumerations in fields such as diagnostics and cell therapy productions.

Acknowledgements

Firstly, I would like to thank my supervisors: Dr Jon Petzing and Prof Rob Thomas at Loughborough University; and Prof Julian Braybrook and Dr Jonathan Campbell at the National Measurement Laboratory. Big thanks especially to Jon, who gave awesome levels of support and guidance throughout this project, who taught me many things, and who pushed me to achieve things I had not dreamt of – I am truly grateful for the experience of working with you. Thank you to Rob for the thoughtful discussions and impactful advice. Thank you to Julian and Jonathan for their expert measurement insights, and for giving me their comments and suggestions on all my research outputs during the PhD.

I wish to thank Liam Whitby at UK NEQAS-LI in Sheffield, whose expertise and insight in the world of clinical cytometry was invaluable in this research project, and whose generosity and enthusiasm made it a delight to collaborate with.

This work would not have been possible without support from the EPSRC Centre for Doctoral Training in Regenerative Medicine at Loughborough University, and LGC. I thank both for the opportunity to undertake such fascinating research.

I would like to acknowledge Dr Peter Kinnell and Dr Karen Coopman, my internal reviewers during the first- and second-year progression reviews, respectively; and credit is due to Dr Rebecca Grant for supervising the initial spark of the CDT mini-project which led on to this thesis.

My sincere thanks go to Dr Shiqiu Xiong at LGC, Fordham, for flow cytometry datasets and fruitful scientific discussions, which were greatly appreciated.

Thank you to all my colleagues at the Centre for Biological Engineering. Special mentions to Alexandros, Hanif, Jon, Preeti, and Ria for providing many moments of welcome distractions from work, and most of all, to Cathy and Maria for their friendship and moral support.

Huge thanks to my fellow CDT students: Bridie, Laurissa, Mateus and Mitchell. I will keep fond memories of the times we spent together and the laughs we had along our PhD journeys.

Part of this PhD was carried out during the height of the Covid-19 pandemic, when the whole country spent months in lockdown. I would like to thank my lifelong friends Anthony, Jenny, Lenny and Siobhan for making those isolation periods bearable and fun with regular Zoom video chats which occasionally descended into musical madness.

Finally, with all my heart I thank my wonderful family. To my dearest parents, David and Josephine, my sister Amanda and my brother-in-law Owen (& Master Theodore), thank you for always being there for me, and for giving me your constant love, support and encouragement – I feel so lucky to have you around me and I love you all.

Publications and Presentations

Publications

- **Cheung M**, Campbell JJ, Whitby L, Thomas RJ, Braybrook J, Petzing J. Current trends in flow cytometry automated data analysis software. Cytometry Part A. 2021; 99(10): 1007– 1021. https://doi.org/10.1002/cyto.a.24320
- **Cheung M**, Campbell JJ, Thomas RJ, Braybrook J, Petzing J. Systematic design, generation, and application of synthetic datasets for flow cytometry. PDA Journal of Pharmaceutical Science and Technology. 2022; 76(3): 1-16. https://doi.org/10.5731/pdajpst.2021.012659
- **Cheung M**, Campbell JJ, Thomas RJ, Braybrook J, Petzing J. Assessment of automated flow cytometry data analysis tools within cell and gene therapy manufacturing. International Journal of Molecular Sciences. 2022; 23(6):3224. https://doi.org/10.3390/ijms23063224
- **Cheung M**, Campbell JJ, Thomas RJ, Braybrook J, Petzing J. Evaluation of flow cytometry automated data analysis platform performance for rare event detection. Cytometry A. 2022. [In preparation]
- **Cheung M**, Campbell JJ, Thomas RJ, Whitby L, Braybrook J, Petzing J. Comparison of manual and automated flow cytometry data analysis methods for CD34+ stem cell enumeration. Cytometry Part B: Clinical Cytometry. 2022. [In preparation]

Oral Presentations

- **Cheung M**, Campbell JJ, Thomas RJ, Braybrook J, Petzing J. Evaluating flow cytometry automated data analysis software in cell therapy manufacturing. FIRM Symposium; 2019 September 23-26; Costa Brava, Spain.
- **Cheung M**, Campbell JJ, Thomas RJ, Braybrook J, Petzing J. Evaluating flow cytometry automated data analysis software in cell therapy manufacturing. 16th Annual bioProcessUK Conference; 2019 November 26-28; Liverpool, UK.

Selected Poster Presentations

- **Cheung M**, Campbell JJ, Thomas RJ, Braybrook J, Petzing J. Flow cytometry automated data analysis software. EPSRC and MRC CDTs in Tissue Engineering and Regenerative Medicine 2019 Joint Conference; 2019 July 12; Manchester, UK.
- **Cheung M**, Campbell JJ, Braybrook J, Thomas R, Petzing J. Benchmarking automated flow cytometry data analysis software using synthetic datasets. Cytotherapy. 2020 May 1;22(5):S38-9. https://doi.org/10.1016/j.jcyt.2020.03.035
- **Cheung M**, Campbell JJ, Thomas RJ, Braybrook J, Petzing J. Evaluation of flow cytometry automated data analysis software performance for rare event detection. CYTO Virtual 2020 (Online); 2020 August 4-5.
- **Cheung M**, Campbell JJ, Thomas RJ, Braybrook J, Petzing J. Assessment of computational tools for automated flow cytometry data analysis within cell and gene therapy manufacturing. Future Leaders in Regenerative Medicine: Joint CDT and UKSB Conference (Online); 2021 June 15-17. **Winner of best poster prize, runner up.**
- **Cheung M**, Campbell JJ, Thomas RJ, Moore P, Whitby L, Braybrook J, Petzing J. Comparability between manual and automated methods for flow cytometry data analysis. 18th Annual bioProcessUK Conference; 2021 November 23-25; Cardiff, UK.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Flow cytometry is a single-cell analytical technique used to characterise and measure cells within academic, clinical and manufacturing laboratory settings. The identification of cell populations in flow cytometry data, typically performed manually, is a significant potential source of variation that can affect the quality of results [1, 2, 3], with incorrect interpretations of the data potentially impacting, for instance, the diagnosis and monitoring of leukaemia and lymphoma [4, 5], and the efficacy of cell therapy products given to patients.

Computational tools that provide automated analysis of flow cytometry data have been developed in recent years to aid the increasingly complex nature of multi-parametric single cell analyses, as well as to improve the reproducibility of results [6]. These automated advances have subsequently opened up questions on how to quantify the uncertainty of cell measurements arising from usage of these software tools. The intention of this thesis is to explore this question within the fields of clinical laboratory analysis and cell therapy manufacturing, in order to understand the requirements for process improvement and to ensure confidence in diagnostics and in the quality of the final cell therapy product that is delivered to patients.

This research aims:

- To examine the variability of automated analyses, both between different software tools and also in comparison to established manual methods.
- To consider whether synthetic datasets can serve as digital reference materials to benchmark automated flow cytometry data analysis software tools.
- To enable the systematic comparison of cell characterisation performances between the various tools.

In this chapter, the cell therapy manufacturing context of the research is introduced,

and a background on flow cytometry is given. The latter part of this chapter presents the research aims and objectives, and ends with an outline of the thesis structure.

1.1 Cell therapy

Cell therapy is the use of living cells that have desired regenerative or curative properties to repair damaged tissue and treat disease. Gene therapy is a related treatment where a patient's cells are modified with the introduction of genetic material. The DNA or RNA genetic material is usually inserted into targeted cells through viral vectors, and can be done either *in vivo* or *ex vivo*. Within this thesis, the term 'cell therapy' is used to cover both cell and gene therapy approaches that produce cells as medicinal products, because both share similar cell manufacture processes and likewise challenges in cell product characterisations.

Cell therapies can be categorised as two types: autologous and allogenic. An autologous therapy is one where the donor is the same as the recipient, so a patient's cells are returned to their own body. In contrast, allogenic therapy involves a different donor to the recipient. Both approaches have their own risks and benefits, such as the low risk of immune rejection but high cost of goods for autologous therapies, versus a higher risk of immune rejection but more attractive 'off-the-shelf' business model for allogenic therapies [7].

1.1.1 Prominent cell therapies and clinical landscape

Prominent examples of cell therapies include haematopoietic stem cell transplantation (HSCT), mesenchymal stromal cell (MSC) therapy, and chimeric antigen receptor (CAR)-T cell cancer immunotherapy.

1.1.1.1 Haematopoietic stem cells

Typically in the case of HSCT, stem cells that have self-renewal and proliferative capacity are harvested from the bone marrow, peripheral blood or cord blood of an autologous patient or an allogenic healthy donor, and used to treat haematological malignancies such as leukaemia. In successful engraftments, the transplanted stem cells reconstitute the bone marrow, where the patient's own cells had previously been destroyed from the chemotherapy or radiotherapy regimens [8].

First performed in humans over 60 years ago with initial unsuccessful results [9], HSCTs have advanced to become a routine treatment with an estimated 1.5 million trans-

plants performed to date in over 1,500 transplant centres worldwide [10]. The reported number of HSCTs in Europe has expanded 10-fold within 30 years (from 4,000 in 1990 to 48,000 in 2019) [11]. Although graft-versus-host disease (GvHD) remains an ongoing challenge for allogenic therapies.

1.1.1.2 Mesenchymal stromal cells

MSCs have also been extensively investigated as cell therapy products for a wide range of clinical indications, due to their accessibility from multiple tissue types (e.g. bone marrow, adipose tissue and neonatal tissues) and their multipotency and immunomodulatory properties [12]. However, there have been limited successes with these MSC-based therapeutics, with most failing to deliver clinically meaningful results as beset by challenges in manufacturing (in particular, the characterisation of an inherently heterogenous population of cells), cell administration to target tissues, and host factors [13].

1.1.1.3 CAR-T cells

In more recent technology advances, CAR-T cells are T cells that have been genetically engineered to express a synthetic receptor composed of an extracellular antigen-binding domain linked to one or more intracellular T cell receptor (TCR) signalling domains, which enables the T cell to specifically recognise cancer antigens on malignant cells and activate an immune response to kill them [14].

CAR-T cell therapies in clinical studies have shown high response rates and remarkable cases of long term remissions in patients with B cell malignancies [15, 16], with notable therapies tisagenlecleucel (Kymriah, Novartis) and axicabtagene ciloleucel (Yescarta, Kite) being approved by the FDA in 2017 and 2018, respectively.

The CAR-T cell field continues to expand, with a further three products since being granted FDA approval: brexucabtagene autoleucel (Tecartus, Kite) in 2020, and idecabtagene vicleucel (Abecma, Celgene) and lisocabtagene maraleucel (Breyanzi, Juno) in 2021 [17]. Reports on the current CAR-T cell therapy landscape identified over 900 products in development from preclinical through to pre-registration stages [18].

In light of these developments, the outlook for the cell therapy market is promising, with estimates of the industry being worth between £9 billion to £14 billion by 2025 [19]. Yet, encouraging results from small scale clinical studies can be a challenge to transfer onto large scale operations intended to widen patient access as well as achieve commercial viability. Lessons from previous commercially released products have identified numerous

challenges faced by cell therapy companies, including manufacturing, delivery, regulatory, and reimbursement hurdles [20].

1.1.2 Cell therapy manufacture

The manufacture of advanced cell therapies broadly involves cell harvest from a donor, followed by the processing, manipulation and culture of the cells in a laboratory, and their administration back into a patient as a biological therapeutic agent [21].

The manufacture process is related but distinct from organ donation or routinely performed HSCTs, where the tissue is stored temporarily outside of the body but not manipulated or cultured (although HSCs form the starting materials of many advanced therapy medicinal products). Manipulation can range from cell selection, cell expansion and genetic modification, but importantly, the characteristics of the resulting cells are significantly altered from the original donor material, and so fall outside of regulatory criteria for "minimally manipulated" cells [22]. The process also has major differences to that of traditional biologics (for instance, monoclonal antibodies, cytokines and inhibitors), because while the cells used in biologics production to synthesise drug substances are considered waste at the end of the process, for cell therapies, the cells themselves are the final product.

Important considerations in the large scale production of these therapies are maintaining the efficacy and safety of the cells while optimising product quality and reproducibility through control of or accommodation of process variation. This task is especially difficult when dealing with living cells and in biological processes where previous research have shown variation up to four orders of magnitude [23].

To understand the process and the sources of variation, analytical methods need to be effective at providing cell characterisation data with high levels of confidence. Critical quality attributes (CQAs) measured during manufacture for in-process testing and end product release testing include identity, purity, potency, viability and safety. Table 1.1 gives some examples of the CQA specifications for CAR-T cell products from previous manufacturing runs [24, 25, 26, 27]. As shown in this table, flow cytometry is heavily used to characterise these CQAs and is therefore a crucial measurement system in cell therapy manufacturing. To that end, validation of the flow cytometry analytical technique is essential to obtain good quality data on the biological product.

Table 1.1: Example critical quality attributes for CAR-T cell products based on [24, 25, 26, 27].

Critical quality attribute	**Test**	**Method**	**Specification**
Identity	CD3+ T cells	Flow cytometry	$\geq$95%
	T cell subsets	Flow cytometry	For information purposes, not criteria for release
Purity	Contaminating cells	Flow cytometry	$<$5%
Potency	Transduction efficiency	Flow cytometry	$\geq$10% CAR+ T cells
	In vitro cytotoxicity	Chromium (^{51}Cr)-release assay	Cell-mediated cytotoxicity
	Cytokine secretion	IFN-γ release assay	IFN-γ production
Sterility	Bacterial and fungal contamination	Automated blood culture system (Bactec), Gram stain	Sterile
Safety	Endotoxin	LAL assay	$<$5 EU/kg
	Vector copy number	PCR	$<$4 copies/cell
	Mycoplasma	PCR	Negative
	RCR/RCL	PCR	No detection
Viability	Trypan blue exclusion	Microscopy	$\geq$70%
	7-AAD staining	Flow cytometry	$\geq$70%
	AO and PI staining	Automatic cell counting	$\geq$70%

1.1.3 Quality landscape in cell therapy manufacturing

Similar to other manufacturing industries, cell therapy manufacturing by pharmaceutical and biotechnology companies adhere to the principles of quality manufacturing that are essential for making consistent and successful products.

Quality guidance documents are set out by global and national regulators. Specifically, the International Council for Harmonisation (ICH) provides guidance on a Quality by Design (QbD) approach in design of manufacturing process, quality risk management and a pharmaceutical quality system for continuous improvement (ICH Q8/Q9/Q10) [28, 29, 30].

These quality guidelines are often adopted by regulatory authorities of individual countries who have the power to approve clinical trials and grant marketing authorisation. For instance, the European Medicines Agency (EMA), the UK Medicines and Healthcare products Regulatory Agency (MHRA) and the US Food and Drug Administration (FDA) produce documents based on ICH guidelines on the quality of a cell therapy drug product and the requirements for investigational advanced therapy medicinal products (ATMP) / Investigational New Drug (IND) applications [31, 32].

1.2 Flow cytometry

Flow cytometry is a technique used to quantify the number of cells in a sample in suspension, and to analyse different cell sub-populations based on the physical characteristics and expression of fluorescently-tagged biomarkers on individual cells.

Flow cytometry is routinely used in diagnostic laboratories for a wide range of clinical applications such as immunophenotyping, DNA content analysis, and quantification of soluble antigens or antibodies [33]. In cell therapy manufacturing, flow cytometry is used to measure key characteristics (e.g. identity, purity, potency and viability) during in-process testing and final product release testing, as mentioned in Section 1.1.2.

Practically, flow cytometry experiments involve staining cell samples with fluorescently conjugated antibodies or other fluorescent dyes, which bind specifically to intra and extracellular target proteins [34]. Single-cell suspensions are required, so solid tissue or adherent cell cultures must be disaggregated or disassociated into single cells through mechanical or enzymatic means prior to flow cytometric analysis. The single cells are flowed in a fluid stream through lasers in a flow cytometer. As the cell passes through the laser beam, light scattering occurs at various angles with forward scatter (FSC) indicative of cell size and side scatter (SSC) proportional to cell granularity and surface roughness.

The fluorescent light emitted by excitation of fluorophores are captured by a system of optics, filters, and detectors. These signals allow specific characterisations of different cell types.

The power of flow cytometry lies in its high throughput capacity to detect thousands of cells per second as well as the use of multiple antibody-fluorophores that can be bound on a single cell and analysed simultaneously. Advances in instrumentation (electronics, detectors and lasers) and reagents (availability of organic and inorganic fluorochromes) have lead to an "explosion" in the number of measurable parameters, from five colours in the 1990s to up to 28-colour panels in 2018 [35, 36]. It is noted that other cytometry platforms such as full spectrum flow cytometry and mass cytometry are capable of analysing over 40 cellular parameters [37, 38], but are not included in this research because of their limited commercial availability and application in cell therapy manufacturing.

1.2.1 Data analysis

The ever-increasing number of parameters used in flow cytometry generates massive complex, multi-dimensional datasets that are increasingly challenging and time-consuming to analyse by traditional, manual methods. The information on hundreds and thousands of single cells from flow cytometry data can be visualised as histograms, or more commonly as two-dimensional (2D) dot plots with different light scatter or fluorescent parameters along the axes.

1.2.1.1 Manual gating

Sub-populations of cells with similar physical or fluorescent characteristics appear as clusters in 2D plots, and in conventional data analysis, these cell subsets are manually selected for further analysis by the drawing of gates in software such as FlowJo (Becton, Dickinson and Company), as shown in the example in Figure 1.1.

A gating strategy involves specific protocols for sequential selection of cell populations. For example, the International Society for Hematotherapy and Graft Engineering (ISHAGE) gating protocol describes the enumeration of CD34+ stem cells from sequential gates that identify lymphocytes, exclude dead cells and debris, and include viable cells [39, 40]. However, despite having highly standardised protocols for reproducible data analysis, a study found that approximately 43% of laboratories who claimed to use the ISHAGE protocol failed to apply it correctly [41].

Gating relies heavily on operator judgement and as a result has limitations in reproducibility, is liable to bias, and results can vary from operator to operator. Previous

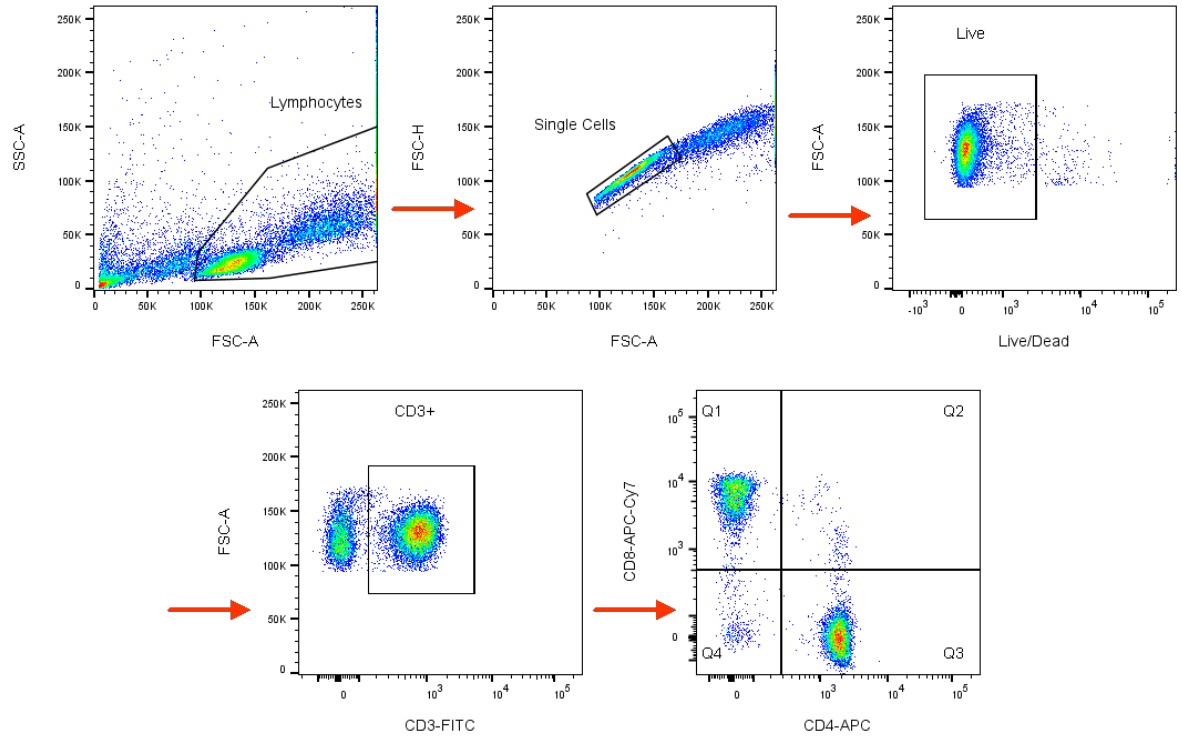


Figure 1.1: Example sequential gating strategy for identifying lymphocyte subsets from peripheral blood mononulear cells (PBMCs). Starting from the forward scatter and side scatter plot, lymphocytes are manually gated, followed by single cells, live cells, CD3+ lymphocytes and finally the CD4+ and CD8+ sub-populations in a quadrant gate.

work has specifically quantified human operator variation when manually gating flow cytometry data that deals with the escalation of gating complexity, and the consequential deterioration of repeatability and increase in operator variation [42, 43].

1.2.1.2 Automated data analysis

To overcome the limitations of manual gating, automated data analysis approaches have been developed in recent years that implement various software algorithms to identify cell populations [44]. These include unsupervised techniques for dimensionality reduction (e.g. t-SNE, UMAP), clustering (e.g. FlowSOM, SPADE, SWIFT), scaffold maps, trajectory inference, and supervised learning methods for classification and regression (reviewed in [45, 46, 6, 47, 48]). These automated data analysis tools are explored in further detail in the literature review in Chapter 2 of this thesis.

The expansion of the field of computational flow cytometry has lead to a disjointed landscape, where tools emerging from different groups are often:

- Implemented in different software/ programming environments.
- Developed using different mathematical approaches to gate cells.
- Tested using distinct datasets.
- Evaluated using different sets of metrics to assess performance.
- Compared with inconsistent selections of other state-of-the-art algorithms.

This apparent lack of standardisation in the field has caused a need for objective testing of the available tools, in order to guide users in choosing the most appropriate software for their analysis purposes.

The Flow Cytometry: Critical Assessment of Population Identification Methods (FlowCAP) consortium organised several competitions to compare the performance of different automated approaches in cell population identification and sample classification tasks [49]. Five different datasets from experiments done on different disease indications and cell types and were used for the population identification challenges (graft-versus-host disease (GvHD), diffuse large B-cell lymphoma (DLBCL), symptomatic West Nile virus (WNV), normal donors (ND), and mouse HSCT). The results of these studies suggested that certain automated methods were able to match manual gating results for cell populations that were relatively easy to identify (being large in size and distinct), however, other populations that were consistently identified manually were missed by algorithms — the reason being that prior biological knowledge on the characteristics of the cells was used by human experts to perform partitioning of clusters.

A separate study to benchmark several automated analysis methods based on accuracy in reproducing manual gating, detection of rare cell populations and recorded runtimes identified FlowSOM, X-shift, PhenoGraph, Rclusterpp, and flowMeans among the high-performing methods tested [47]. Six high-dimensional datasets from experiments in immunology were used in these comparison studies, plus the ND and WNV datasets from FlowCAP studies.

1.2.1.3 Implications of inaccurate data analysis

The implications of inaccurate manual or automated analysis of flow cytometry data are potentially huge in terms of financial, time, labour, health and emotional costs.

For instance, in the context of diagnostics for haematological cancers, a patient sample that is absent of disease but incorrectly reported as disease positive (false positive) may be referred to undergo unnecessary treatment with associated medical costs (for the National Health Service (NHS)/ insurance companies/ the individual) and emotional stress to the patient and their families. Whereas the inverse case of a patient with disease, whose

sample is misdiagnosed as disease free (false negative), would miss out on the benefits of early treatment and potentially have a worse clinical outcome when their disease is eventually detected at a later time point.

In another example in the context of cell therapy manufacturing, inaccurate analysis of a product's CQAs (e.g. specification states a viability of $\geq$70% to pass but analysis under-reports this to 68%) could result in a product being stopped in-manufacture because of the risk of the completed out-of-specification batch not being certified and released for patient use by the Qualified Person.

The consequences of this could mean the need to restart the manufacturing run, causing a setback of several weeks, plus additional costs of goods, staff, facility and equipment requirements. Most significant of all is the delay to patients receiving life-saving treatment. For an autologous therapy this would impact one patient, but for an allogenic therapy potentially hundreds of patients would be affected.

1.2.2 Efforts to reduce the variation in flow cytometry

The flow cytometry community have long recognised the need for standardisation and importance in reproducible results. With respect to data, the International Society for Advancement of Cytometry (ISAC) have coordinated initiatives in data standards such as MIFlowCyt [50] to outline the minimum information about a flow cytometry experiment that should be reported, the FCS data file format standard [51], and the markup language gating-ML [52] for reproducible gating data analysis compatible with different software tools.

Efforts to reduce the variation in data analysis of particular cell populations have been led by specialists in the field. For example, immunologists from the EuroFlow consortium [53] and Human ImmunoPhenotyping consortium [54] aim to standardise cytometer settings, antibody panels, sample preparation SOPs, gating strategies and data analysis tools for immunophenotyping of leukocytes. These consortia performed multi-centre studies to compare variation in results after implementation of standards.

Much of the work to standardise flow cytometry assays stems from clinical laboratories. Clinical flow cytometry is used to diagnose or monitor disease, and it is important that the same diagnostic test produces consistent results from different laboratories. Guidelines on quality and best practice reference documentation are available from the International Clinical Cytometry Society (ICCS) Quality and Standards Committee [55], and the Clinical and Laboratory Standards Institute (CLSI) [56].

Proficiency testing schemes such as those run by UK National External Quality As-

sessment Scheme for Leucocyte Immunophenotyping (NEQAS-LI) aim to monitor performance from clinical laboratories [57]. Reference material is sent to participating centres for analysis, then results are aggregated to evaluate technical variability and identify laboratories that falls outside the agreed range limits. However, the method of data analysis is not controlled because laboratories apply their normal protocols to analyse the material. These studies do not look specifically at the effects of automated data analysis on a result.

The National Institute of Standards and Technology (NIST) runs a quantitative flow cytometry measurement program which aims to develop tools for measurement assurance such as reference materials, process controls and performance specifications [58]. Currently, none of these quality assessment schemes evaluate flow cytometry data analysis using automated methods.

1.3 Problem statement

Addressing the variation arising from manual analysis (gating) of flow cytometry data is critical to achieving confidence in cell measurements, and numerous computational tools have been developed in recent years to improve reproducibility, and bring an objective and unbiased approach to the gating process. Previous comparison studies that critically assessed these automated gating methods showed their development has advanced to the point where they are able to replicate some manual cell population identification [49], and several high performing tools have been highlighted [47].

However, these comparison studies have relied on using a narrow range of manually gated experimental datasets as the reference materials to benchmark computational tools. This is clearly problematic, firstly, because of the human subjectivity and bias inherent in manually estimated reference cell count values. Secondly, the lack of 'ground truth' absolute cell counts available to validate computational tools against means that explicit statements of measurement accuracy and repeatability cannot be provided. This gap in the knowledge of the uncertainty of measurements from automated tools leads to a risk where users accept the reported values as true, without understanding the accuracy and repeatability of the outputs, or how it was obtained.

Therefore, developing alternative, robust datasets to enable systematic and comprehensive comparisons between different automated tools is desirable to provide further assurances in cell characterisations and build trust and transparency in automated tools among the community. Having high quality flow cytometry data could impact, for example, the diagnosis of haematological malignancies in patients and the decisions on their

treatment, or the release of cell therapy products that fall out of specification from manufacturing sites.

1.4 Research scope

This research is focussed on variation in the data analysis part of the flow cytometry process, downstream of data acquisition. Upstream sources of variation will not be considered, for instance, raw materials (e.g. antibodies, dyes), sample processing steps (e.g. lysis, wash, incubation times) and equipment (e.g. cytometer instrument and settings). Although these factors all contribute to the overall reproducibility of flow cytometry, each would require separate comparison studies centred around those factors as variables.

The wide array of computational tools available necessitated the narrowing of the research scope to unsupervised learning algorithms specifically applicable to flow cytometry, with the exclusion of other machine learning techniques such as supervised and semi-supervised learning algorithms, and those developed for other analytical techniques, such as mass cytometry or single-cell RNA sequencing (scRNA-seq).

Additionally, since this research intends to develop synthetic datasets for benchmarking software tools, a further justification for the exclusion of supervised learning techniques lies in the significantly different approach required in actual methodology of synthetic dataset design and application, mainly the need for substantial amounts of labelled training and testing datasets.

The performance of different computational tools can be evaluated based on a number of metrics. This thesis considers the performance related to cell population identification accuracy and repeatability metrics; factors such as run times of software (fundamentally hardware dependent) and their ease of use have arguably less significance to the quality of data, and lie outside the scope of this research.

In addition, a large component of this research is the comparison of different automated techniques across different mathematical operators. It is important to state here that the goal is not to develop a new algorithm to improve on the ones currently available, nor to drive one particular technique to its optimal performance.

1.5 Aims and objectives

The overall aim of this research is to define the confidence of flow cytometry automated data analysis software tools. This can be broken down into the following research objectives and corresponding questions:

1. To understand the landscape of automated flow cytometry data analysis software in the context of academic, clinical and manufacturing laboratories.

    - What automated software tools are available?
    - How do the software tools differ?
    - How widely used are the software tools?
    - How do their usages differ between academic, clinical and manufacturing settings?

2. To explore benchmarking datasets applicable for the critical assessment of automated flow cytometry data analysis software.

    - What properties of flow cytometry benchmarking datasets are required for testing of software tools?
    - Can synthetic datasets be designed and generated with these properties?
    - What are the advantages and disadvantages of synthetic datasets compared with real-world datasets?

3. To compare the performances of different software tools in cell population identification tasks.

    - What is the effect of varying the distance between clusters on software performance (in terms of accuracy and repeatability)?
    - Are certain software more sensitive to non-normal cluster distributions?
    - What are the limits of detection of the different software when challenged with a rare cell dataset?
    - How robust are software tools to noise elements in the data?

4. To analyse the variation between automated software outputs in comparison to manually analysed data.

    - How does the variation compare between manual and automated software analysis of flow cytometry data?
    - Does the variation differ when analysing cell populations with different degrees of separation?
    - What is the correlation between the two analysis methods?

1.6 Novelty

Based on the gap in the current research identified in the problem statement above, it is anticipated that the novelty of this thesis lies in the development of a range of synthetic datasets with controlled data properties, along with their application to systematically evaluate the cell population identification performances of different automated data analysis software. This approach is different to previous comparison studies of computational tools for flow cytometry, none of which have used synthetic datasets, giving the novel approach here the benefit of having a known 'ground truth' reference value to directly and absolutely compare software outputs against, and objective statements of accuracy and repeatability.

1.7 Thesis structure

This thesis is structured as follows:

Chapter 1 - Introduction

This chapter introduces the background of automated flow cytometry data analysis and the context of the cell therapy field, and sets out the research aims and objectives.

Chapter 2 - Literature review

This chapter provides a comprehensive literature review on the current landscape of computational tools, paired with a survey of their usage among clinical laboratories, to enable selection of a range of different software tools that are relevant and representative of the field for subsequent comparison studies in Chapters 4 to 7.

Chapter 3 - Synthetic datasets

This chapter explores the common characteristics of flow cytometry datasets and demonstrates an approach to systematically design and generate synthetic datasets with relevant characteristics in a highly controlled manner.

Chapter 4 - Software comparison

This chapter applies synthetic datasets with different distances between clusters, and clusters with normal or non-normal (skewed) distributions, to assess the performance of automated software.

Chapter 5 - Rare cells

This chapter focuses on rare cell detection, which is an important component of flow cytometry, and applies rare cell synthetic datasets to understand the limits of detection of different software. The results of the comparison study are validated using real experimental datasets containing rare cell populations.

Chapter 6 - Noise

This chapter investigates the robustness of computational tools to noise elements which are ubiquitous in flow cytometry data. The synthetic datasets from Chapters 4 and 5 are modified with the injection of noise components, and applied to the different software tools.

Chapter 7 - NEQAS

This chapter analyses the variation between automated software outputs compared to manual analysis, using a clinical real-world dataset from UK NEQAS-LI. The results from this chapter provide an understanding of the variation apparent in software in the wider context of the current 'gold standard' of manual gating.

Chapter 8 - Conclusion

This chapter summarises the major research findings from this thesis, outlines the central contributions, and suggests further work towards achieving confidence in automated flow cytometry data analysis techniques.

Chapter 2

Literature review

The publication listed below was an outcome of the work reported in this Chapter:

> **Cheung M**, Campbell JJ, Whitby L, Thomas RJ, Braybrook J, Petzing J. Current trends in flow cytometry automated data analysis software. Cytometry Part A. 2021; 99(10): 1007–1021. https://doi.org/10.1002/cyto.a.24320

2.1 Introduction

Flow cytometry (FC) is an important analytical technique for single-cell population identification and characterisation. It is widely used within biotechnology, pharmaceutical, and, clinical laboratories and biomanufacturing spaces. Reproducibility and rigour in results are very important, driven by the needs of regulators around the world, however, a major source of variation in FC lies within data analysis [1]. Conventional FC data analysis involves sequential manual selection (gating) of regions of interest typically in two-dimensional scatter or contour plots, viewing different combinations of parameters as axes. The analysis is straightforward with three- to four-colour immunofluorescence data but becomes significantly more complex when examining an increasing number of cellular markers, leading to increasing human operator variation and issues of reproducibility [42, 43]. Current state-of-the-art flow cytometers are capable of measuring over 40 parameters, generating challenging complex and time-consuming multidimensional datasets for manual analysis [59, 60, 36].

The past decade has seen a growth in the field of computational FC as researchers become increasingly motivated to solve the process bottlenecks, and, reproducibility issues in manual gating, and improve standardisation in immunophenotyping [61]. New automated data analysis software packages have emerged, making use of a range of differ-

ent machine learning and clustering algorithms to replicate or aid manual data analysis tasks such as; data pre-processing, cell population identification and enumeration, feature extraction and sample classification [44]. Visualisation of data processed through algorithmic analyses is an essential aspect of analysis workflows, and is often embedded in the automated analysis itself, therefore making the distinction between pure analysis tools versus visualisation tools somewhat blurred. Graphical outputs aid quality control checking and enables understanding and interpretation of the data.

Examples of FC visualisations can be: 1) cell populations colour coded according to clustering results displayed on classic biaxial dot plots, 2) grouped populations in nodes arranged in the form of spanning trees, and 3) mapping of high-dimensional data to two-dimensional scatter plots representing data similarities, with colour coded cell clusters. These data-driven automated algorithms have been demonstrated to improve the quality of flow cytometry data compared to centralised manual analysis, with potential benefits in lower technical variability in certain cell populations, reduced bias and better efficiency [54]. Given the proliferation of such algorithms, verification methods to ensure correct choice would be recommended. It would be sensible for all users to contextually develop their own robust testing measures for automated analysis. However, this raises subjectivity issues if testing was based on users own biological knowledge, compounded by the fact that there are no common tool sets to achieve this apart from real-world data sets which do not necessarily have an absolute cell count, and are inflexible compared to the potential of synthetic data.

Typical workflows in computational cytometry can be divided based on tools used for discovery versus targeted analysis, i.e. the detection of unknown, novel cell populations compared with known well-defined ones. In both contexts, automation can help to reduce variability in the data analysis process. In discovery mode, automated tools can help uncover cell populations overlooked in sequential manual gating strategies, such as cells gated out in earlier steps. The value of automated tools in discovery mode is especially notable in facilitating interpretation of high dimensional (>30) data, as the data can be reduced and visualised in two dimensions. These tools assist with the data exploration process, help to give an overview of the structure of the data, identify relationships between variables and offer novel insights. For comparison, in targeted analysis mode, the cell populations of interest are well characterised, the data analysis process follows a standard protocol that is likely to be validated and approved, for example, in clinical flow laboratories carrying out high throughput screening; measurement of clinical trial endpoints for haematological malignancies. The benefit of automated tools here may be in reducing the workload on users by automating classification of healthy or disease cases,

only flagging up uncertain cases for manual interpretation, thereby speeding up the data review process.

As the number of automated software tools increases, comparison studies have become important to provide guidance for users to determine which software tool to use for their analysis, and to evaluate the performance of the software tools. The Flow Cytometry: Critical Assessment of Population Identification Methods (FlowCAP) consortium initiated a series of open challenges to objectively evaluate these new computational methods [49, 62]. FlowCAP provided benchmarking datasets to critically assess performance in population identification and sample classification tasks and used the F-measure (the harmonic mean of precision and recall) to rank the algorithms. These rankings helped inform potential users on the quality of automated methods based on different tasks. FlowCAP demonstrated certain automated methods (such as ADICyt, flowMeans, FLOCK and FLAME) were able to reliably replicate manual gating for some of the datasets used in the challenges.

Several other recent comparison studies have evaluated selected unsupervised clustering methods in their abilities to reproduce manual gating, detect rare cell populations and their runtimes. Among those, one study [47] identified FlowSOM as the best performing clustering method along with the fastest runtimes. X-shift, PhenoGraph, Rclusterpp, and flowMeans were also mentioned to perform well across six high dimensional datasets. A separate study [3] assessed FLOCK, SWIFT, and ReFlow on their ability to detect low-frequency T cell populations compared with central manual gating. SWIFT was found to outperform the others in terms of the identification of populations at frequencies below 0.1%. This study noted the difficulties in implementing a fully automated workflow without human intervention. In addition, one study [63] evaluated the reproducibility and robustness of results based on the cluster stability, with PhenoGraph observed to generate the highest proportion of stable clusters compared with SPADE1 and FLOCK.

Despite these recent benchmarking studies, uptake of automated analysis among academic, biotechnology, pharmaceutical, clinical laboratories and contract research organisational researchers has been slow and manual gating remains the default method and standard. Manual analysis can be performed on instrument-packaged software (e.g. Becton Dickinson FACS Diva, BD FACS Canto, Beckman Coulter Navios) or stand-alone FC analysis software (e.g. FlowJo, FCS Express, Kaluza, VenturiOne). The primary reasons for clinical centres not employing automated analysis was recently cited as being a lack of trust/understanding and lack of resources [64]. In this regard, this novel analysis of automated software provision and use presented here, is intended for researchers and process operators familiar with FC who do not necessarily have a computational background,

who are interested in implementing automated methods into their data analysis workflow and require a better understanding of the opportunities for automated software package selection.

This analysis begins by identifying the most frequently used tools in the past 10 years in FC automated data analysis. Popular software tools are identified based on literature citations, then their common features are outlined to allow the determination of the toolset most relevant to individual need. In addition, automated data analysis adoption trends from front line clinical laboratories are identified through a survey, and insights are provided on the reasons uptake of certain software tools is higher than others.

2.1.1 Chapter aims

The aims of this chapter are to:

- Provide a comprehensive overview of the currently available automated flow cytometry data analysis tools.
- Identify the most popular tools based on literature citations within academia, and survey responses from clinical and biomanufacturing laboratories.
- Classify the different algorithmic approaches implemented by software tools to identify cell populations.
- Understand current challenges faced by users in the application of these automated tools.
- Enable the selection of relevant software for performance comparisons going forward in this thesis.

2.2 Search strategy

The goal of part of this research was to understand current trends in automated data analysis software tools, the characteristics of these tools, and identify which ones were the most popular (although it is recognised that this is not necessarily a measure of most effective software tool). Software tools mentioned in recent reviews [49, 47, 3, 63, 6] were included. In addition, the Web of Science (WoS) database was searched using the following keywords; flow cytometry, automated, analysis. Using this search strategy, 89 software tools were identified from recent reviews and 108 publications were returned from the WoS database, typically output from research, clinical and biomanufacturing facilities. The WoS search strategy was designed to be as comprehensive as possible, although some tools may have been missed due to the fragmented nature of the field,

such as FLOW-MAP force directed graphs [65] and scaffold maps [66]. Use of additional keywords such as 'computational' may have highlighted more software tools, however in practice the records retrieved from the database were either too restrictive with the AND Boolean search operator, or excessively broad with the OR search operator. After removing duplicates, software tools identified in the search were refined based on the following specifications.

Inclusion criteria:

- Software is detailed in a publication from a peer-reviewed journal.
- Publication type: article.
- Software for flow cytometry or mass cytometry.
- Software for automated cell population identification (gating).
- Software intended for identification of human or mammalian cells.
- Software source code is available, or program is made accessible by authors.

Exclusion criteria:

- Software lacking publication from a peer-reviewed journal.
- Publication type: conference proceedings, reviews, editorial material, book chapters. This exclusion criteria was applied in order to capture work that actually applied the data analysis software tool rather than just citing their use.
- Software unrelated to flow cytometry or mass cytometry technique.
- Software solely for automated data pre-processing, compensation, transformation, or other quality control feature.
- Software unrelated to identification of human cells (e.g. beads, phytoplankton, bacterial identification) to focus the scope on cell therapy and medical applications.
- Software source code not provided, or program inaccessible.

Certain proprietary software tools that fell into the exclusion criteria include automated cell identification features in FACS Diva (Becton, Dickinson & Company (BD)), Kaluza (Beckman Coulter), FlowJo (BD), FCS Express (De Novo Software), Gemstone (Verity Software House) and VenturiOne (Applied Cytometry).

The number of software tools matching the criteria was refined to 51. Once shortlisted, software tool popularity was ranked according to the number of article citations. The sum total of the number of citations across all 51 software tools was 2,027. Citing articles were refined to those matching 'cytometry' as a keyword, included articles, and excluded conference proceedings, reviews, editorial material and book chapters. Additional software

tool would have been identified if the search strategy were broadened to include automated single-cell analysis approaches from other technologies (e.g. RNA-sequencing analysis tools in genomics, single-cell imaging, single-cell proteomics), and indeed many tools are transferable between different omics domains, however this was beyond the scope of this work.

2.3 General findings and trends

This literature survey was initially completed at the beginning of this research project, in 2019, and identified 51 automated flow cytometry software tools (Table 2.1). Newer software tools identified from more recent literature searches are highlighted in Section 2.3.3.1, however these tools have not been included in the main analysis reported below due to the work evolution and time limitations of the research project. The earliest software tool was released in 2008 and subsequent years saw the number of different software tools released ranging from 1 to 6 per year, except for 2014 when a peak of 11 software tools were published (Figure 2.1A). When considering country of origin, the USA has led the development with 29 software tools, followed by Canada with 6 software tools. Outside of North America, a small number of European studies have come from The Netherlands, Belgium, France and Germany (4, 2, 2 and 2 software tools respectively). Australia and Singapore have also produced two apiece (Figure 2.1B).

The environment in which users interact with the software tools range from basic command line inputs to full graphical user interfaces (GUI). This survey found 41% of software tools could be accessed with a GUI, compared with 59% without GUIs (Figure 2.1C). A caveat here is that although most likely to have GUIs, as identified in Section 2.2, proprietary computational tools lacking peer-reviewed publications and with unavailable source code were excluded from this survey. Many of the tools were available in multiple programming languages, offering FC analysts a choice of integrated development environments. This survey found 59% of the software tools were available in R, 29% in Matlab and 18% in Python (Figure 2.1D).

Table 2.1: Software identified in literature survey, ranked according to number of citations.

Rank	Software name	No. of citations	Abbreviation	Purpose	Reference
1	viSNE	294	visualization tool for high-dimensional single-cell data based on the t-Distributed Stochastic Neighbor Embedding (t-SNE) algorithm	Visualisation of high-dimensional single-cell data via dimensionality reduction	[67]
2	SPADE	236	Spanning-tree progression analysis of density-normalized events	Visualisation of high-dimensional cytometry data by downsampling, clustering and a minimal spanning tree	[68]
3	t-SNE	194	t-Distributed Stochastic Neighbor Embedding	Dimensionality reduction for visualisation	[69]
4	Pheno-graph	156		Model cellular phenotypes	[70]
5	FLAME	107	Flow analysis with automated multivariate estimation	Identify cell populations by multivariate mixture modelling	[71]
6	Citrus	87	Cluster identification, characterization, and regression	Identification of stratifying cellular subpopulations	[72]
7	FlowSOM	75	Self-organizing map	Clustering data into self-organizing maps and visualisation by minimal spanning trees	[73]
8=	DensVM	70	Density-based clustering aided by support vector machine	Cell population identification and classification	[74]
8=	flowMeans	70		Cell population identification by k-Means based clustering	[75]
10=	ACCENSE	66	Automatic Classification of Cellular Expression by Nonlinear Stochastic Embedding	Identification of cell subpopulations through t-SNE dimensionality reduction and density-based partitioning	[76]
10=	Wanderlust	66		Developmental trajectory detection	[77]
12	FLOCK	63	FLOw Clustering without K	Cell population identification by density-based clustering	[78]
13	flowClust	58		Cell population identification by multivariate t-mixture modelling with Box-Cox transformed data	[79]
14	flowMerge	53		Cell population identification using flowClust and a cluster merging algorithm	[80]
15	X-Shift	50		Exploration of single-cell data by clustering (K-nearest neighbour density estimate) and visualisation by divisive marker trees and force-directed layouts	[81]
16	SamSPEC-TRAL	48		Cell population identification by spectral clustering and sampling	[82]

Table 2.1 – continued from previous page

Rank	Software name	No. of citations	Abbreviation	Purpose	Reference
17=	flowPeaks	42		Cell population identification by K-means clustering and density peak finding	[83]
17=	OpenCyto	42		Mimicking manual gating based on hierarchical automated gating pipelines	[84]
19	Mixture model	41		Cell population identification by mixture modelling	[85]
20	HPDGMM	31	Hierarchical Dirichlet Process Gaussian Mixture Model	Rare event detection and cell subset alignment across multiple samples	[86]
21	flowDensity	26		Mimicking manual gating based on cellular density distributions	[87]
22	SWIFT	24	Scalable Weighted Iterative Flow-clustering Technique	Identification of rare cell populations based on Gaussian mixture model-based clustering	[88, 89]
23	HSNE	16	Hierarchical Stochastic Neighbor Embedding	Visual exploration of the hierarchy in cytometry data	[90]
24	Misty Mountain	15		Cell population identification by density contour clustering	[91]
25=	COMPASS	14	Combinatorial Polyfunctionality Analysis of Single Cells	Identification of cell subsets correlated with clinical outcomes	[92]
25=	FlowFP	14	Fingerprinting for Flow Cytometry	Generation of multivariate distribution 'fingerprints'	[93]
25=	immunoClust	14		Cell population identification by iterative model-based clustering	[94]
28	JCM	12	Joint Clustering and Matching	Cell population identification and matching across a batch of samples	[95]
29	flowType/ RchyOptimyx	11		Cell population identification by partitioning and correlation with clinical outcomes	[96]
30=	ASPIRE	10	Anomalous sample phenotype identification with random effects	Identification of anomalous samples with random effects	[97]
30=	DeepCyTOF	10		Cell classification by deep learning	[98]
32	AutoGate	9		Sequential selection of cell subsets and visualisation	[99]
33	FloReMi	8	Flow Density Survival Regression Using Minimal Feature Redundancy	Survival time prediction	[100]
34	CCAST	7	Clustering, Classification and Sorting Tree	Isolation of homogenous subpopulations	[101]
35	flowLearn	6		Identification and quality checking of cell populations	[102]

Table 2.1 – continued from previous page

Rank	Software name	No. of citations	Abbreviation	Purpose	Reference
36	ACDC	5	Automated Cell-type Discovery and Classification	Cell population discovery and classification	[103]
37=	Competitive SWIFT	4	Scalable Weighted Iterative Flow-clustering Technique	Sample comparison by competitive clustering	[104]
37=	SPADE 3	4	Spanning-tree progression analysis of density-normalized events	Visualisation of high-dimensional cytometry data by downsampling, clustering and a minimal spanning tree	[105]
39=	cytometree	2		Cell population identification based on a binary tree algorithm	[106]
39=	DAFi	2	Directed Automated Filtering and Identification of cell populations	Cell population identification based on recursive data filtering and clustering	[107]
39=	diffcyt	2	Differential discovery in high-dimensional cytometry via high-resolution clustering	Differential discovery analysis	[108]
39=	FlowVIEW	2		Quantification of cell populations via a supervised learning approach	[109]
39=	LDA	2	Linear discriminant analysis	Prediction of cell populations	[110]
44=	ECLIPSE	1	Elimination of Cells Lying in Pattern Similar to Endogeneity	Identification of disease-specific cells	[111]
44=	NPflow	1	Bayesian Nonparametrics for Automatic Gating of Flow-Cytometry Data	Cell population identification by model-based clustering	[112]
44=	PSM with GemStone	1	Probability State Modeling	Cell population identification via a probability-based approach	[113]
44=	SOPHE	1	Second order polynomial histogram estimators	Cell population identification by data binning	[114]
48=	PHATE	0	Potential of heat diffusion for affinity-based transition embedding	Dimensionality reduction for visualisation	[115]
48=	SIC	0	Subset Identification and Characterisation	Subset identification and characterisation pipeline	[116]
48=	SigClust	0	Signature based Single-Cell Clustering	Cell population identification using phenotypic signatures	[117]
48=	UMAP	0	Uniform Manifold Approximation and Projection	Dimensionality reduction for visualisation	[118]

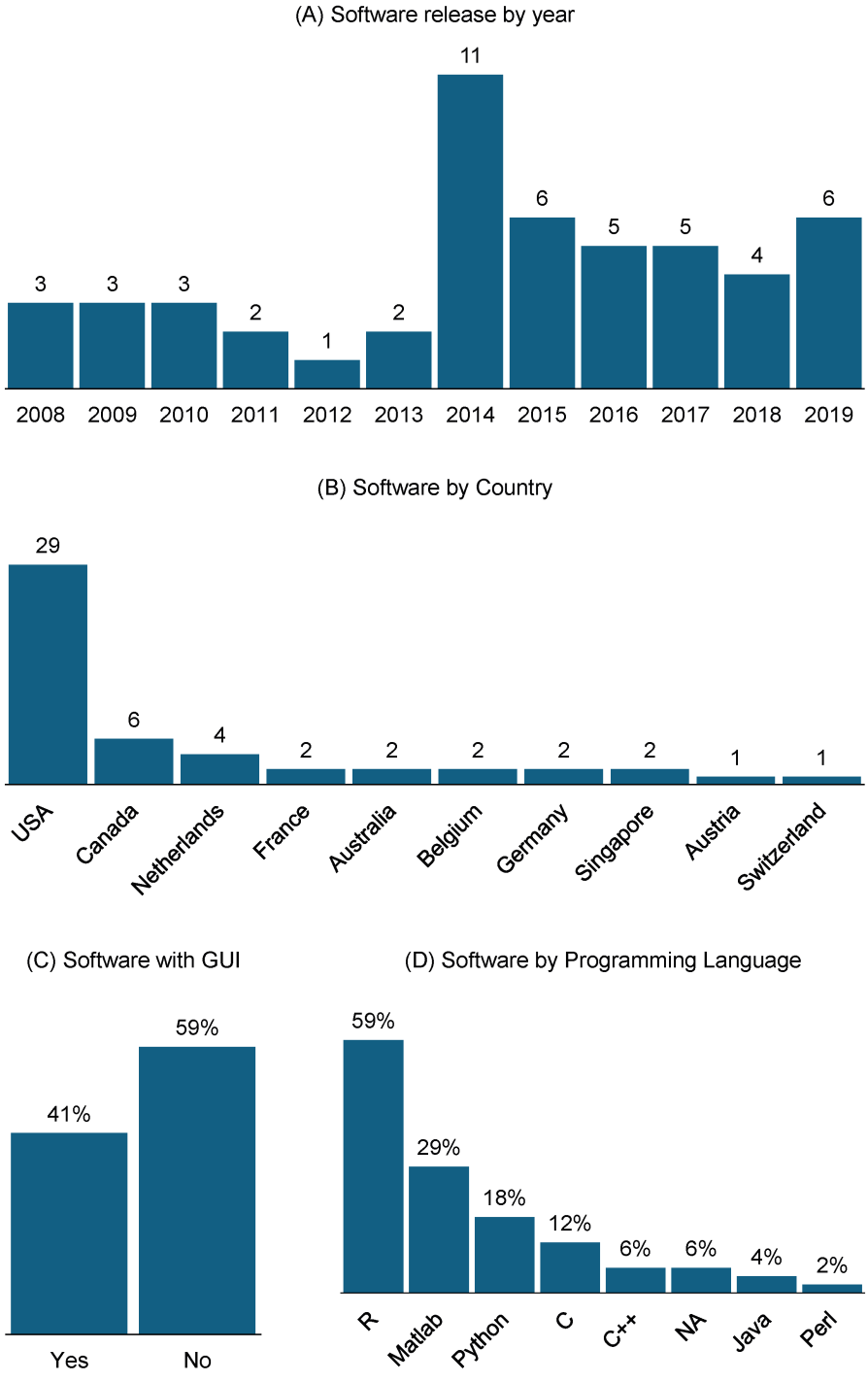


Figure 2.1: General trends in automated data analysis software tools. A) Number of software tools released by year gradually increasing. B) USA leads the development of software tools to analyse flow cytometry data. Counts based on first author affiliation in publications. C) The majority of software tools are released without graphical user interfaces (GUI). D) Technical trends. Software tools are released in multiple programming languages and implementations. R, Matlab and Python are the languages most software tools are available in.

2.3.1 Most used software tools

The findings from the literature survey revealed the top 5 most cited automated data analysis software tools on the basis of the search criteria and exclusion criteria were: viSNE, SPADE1, t-SNE, PhenoGraph and FLAME (Table 2.1). To balance out the effect of earlier software tool releases accumulating more citations over time, the number of citations were averaged over the number years in publication leading to an adjustment of the highest citation rates; viSNE, PhenoGraph, SPADE1, FlowSOM and t-SNE (Figure 2.2). Changes in individual software tool citations over time showed viSNE has been the top cited software tool for the past three consecutive years (Figure 2.3), and a recent rapid increase in FlowSOM citations, moving it from 23rd most cited software tool in 2017 to 7th highest in 2019. viSNE has a higher citation rate than its origin dimensionality reduction method t-SNE, suggesting many authors consider these separate tools and neglect to cite the original van der Maaten publication [69].

Software tools that provided a GUI were considerably more cited than those without – command line-based software tools (Figure 2.4). The combined total number of citations for software tools with GUIs was 1,459 compared with 613 for those without GUIs. Command line-based software tools require computer programming knowledge, which acts as a potential barrier to many biomedical researchers. Another factor that influences software tool selection is cost and availability. There are three broad levels of cost in accessing automated flow cytometry data analysis software tools: free open source software on a free platform, free open source software on a platform requiring a licence fee or subscription, and, commercial software on a standalone or paid platform. Currently, access to software tools are mostly free and open source, however, some platforms require a paid subscription. Software tools are available as packages built within the Matlab or R statistical software environments, plugins as part of specialist FC manual data analysis software (such as FlowJo, FCS Express), and applications on web-based platforms such as Cytobank [119]. The same software tool can be implemented and be available on more than one platform. Cost does not appear to be a deciding factor for users, because the most cited software tools were accessed through paid platforms (Figure 2.5). The levels of usability and software support provided typically increase in line with cost.

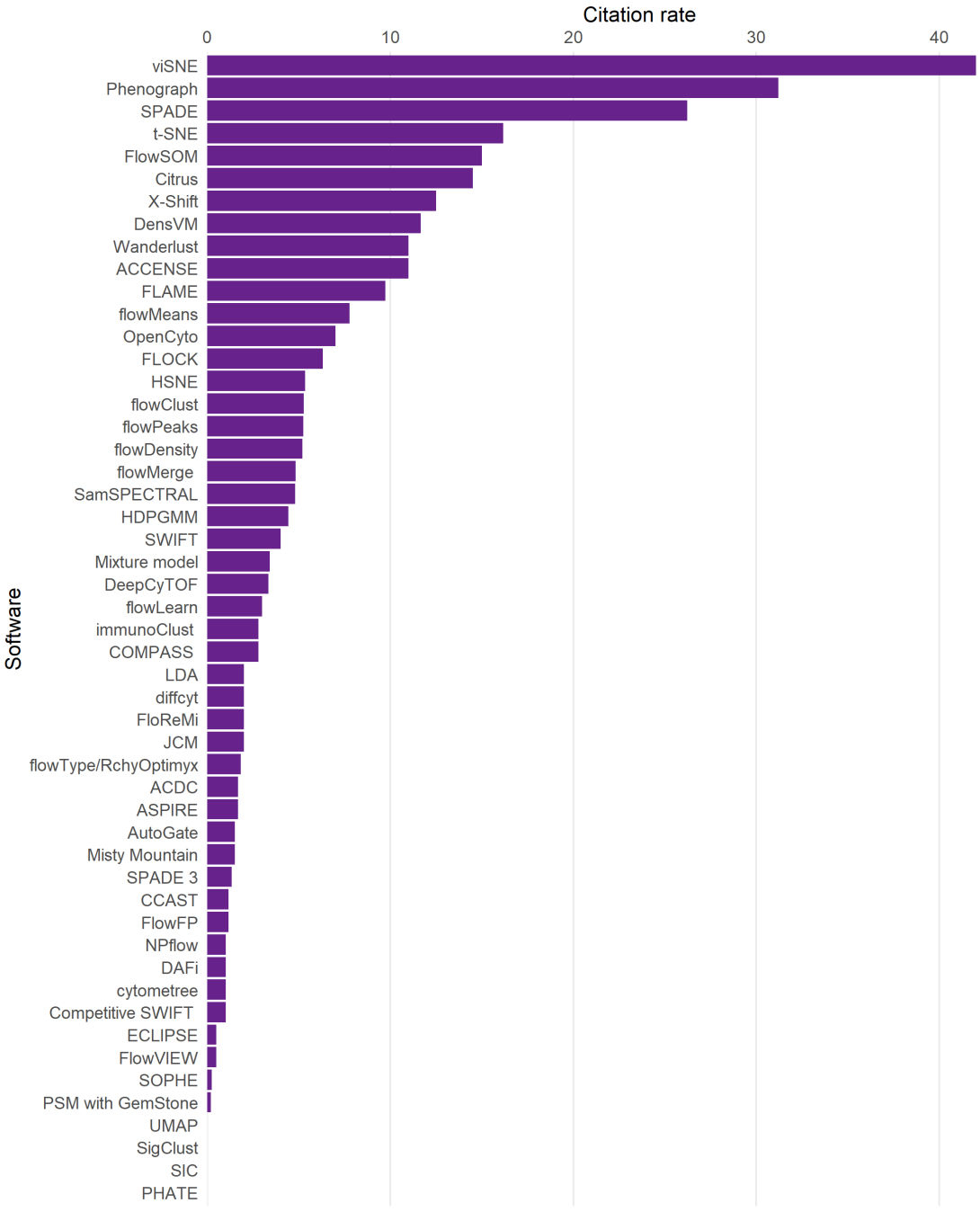


Figure 2.2: Software citation rates.

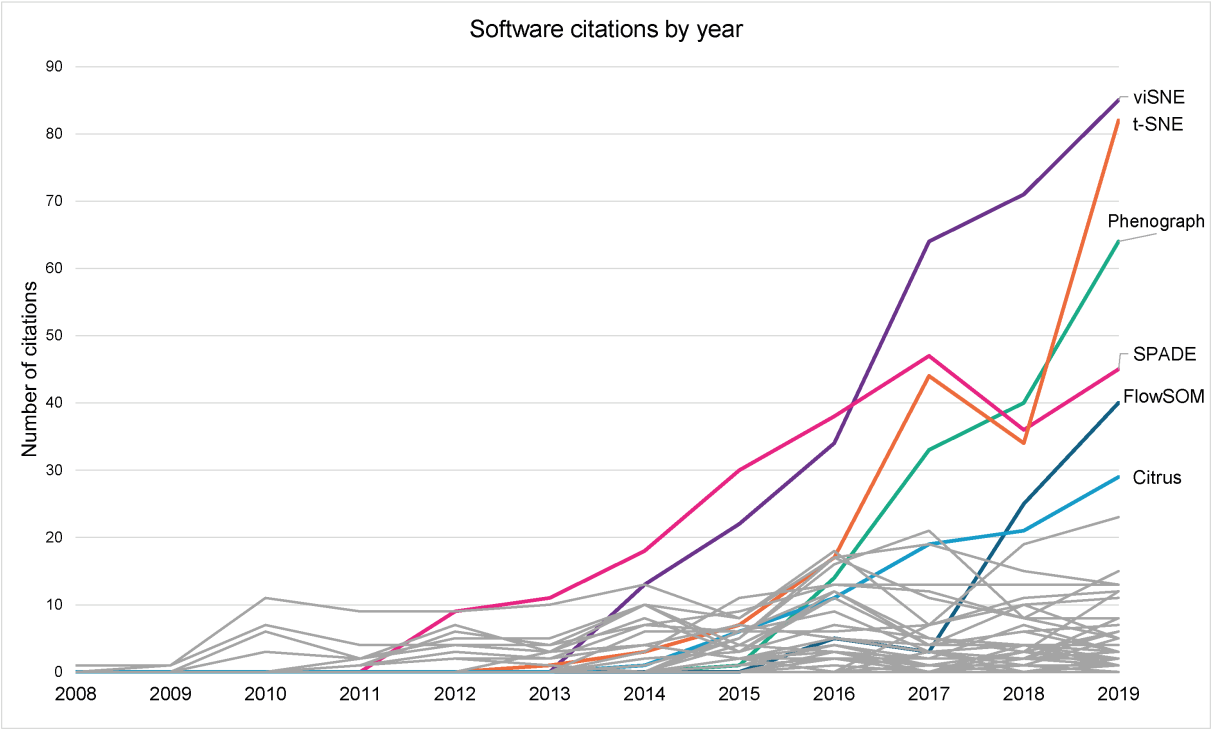


Figure 2.3: Software citation trends by year.

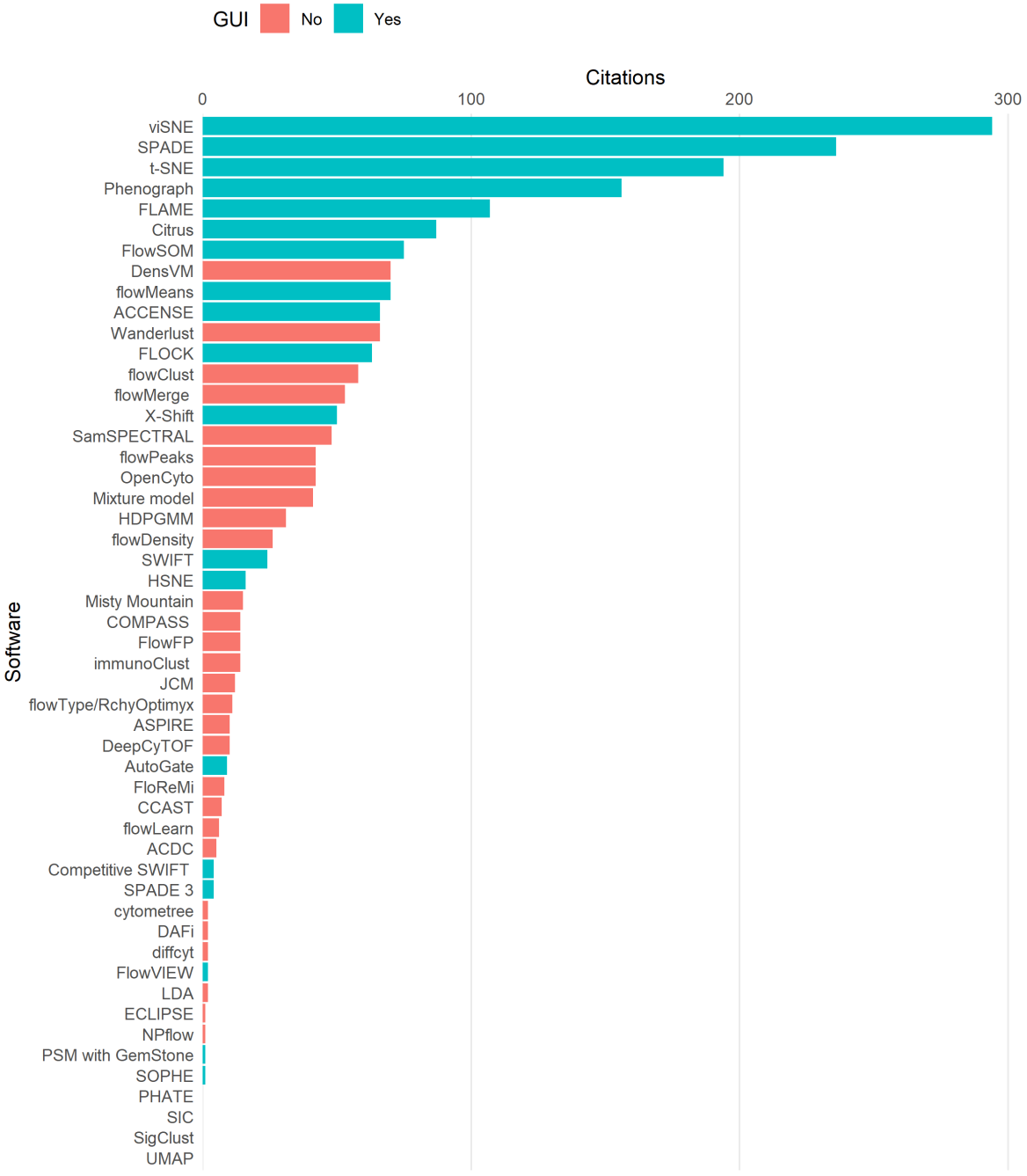


Figure 2.4: Number of citations by presence of graphical user interfaces (GUIs)

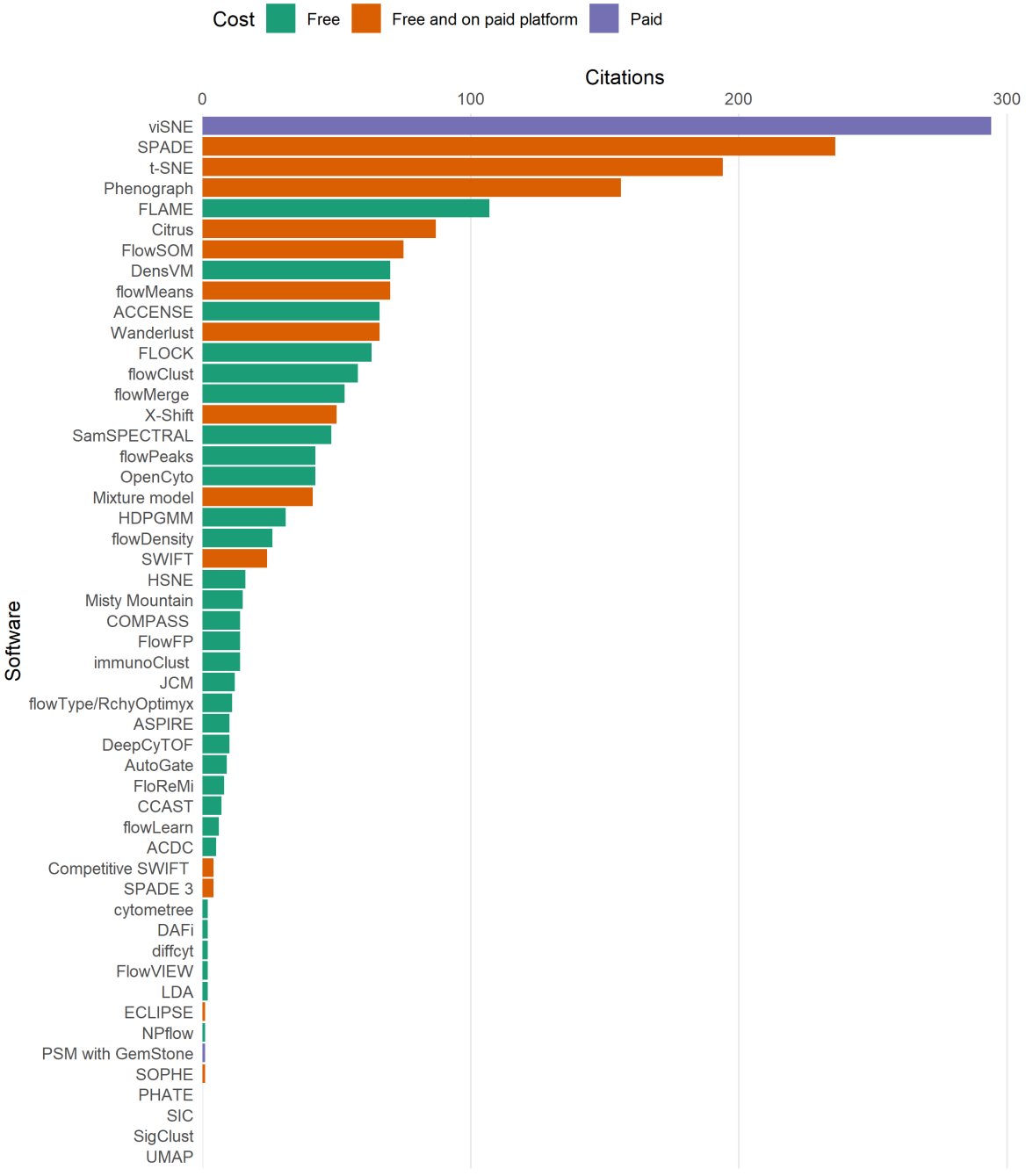


Figure 2.5: Number of citations based on cost of software tool.

2.3.2 Software algorithm types

For further insight, the software tools were separated based on the algorithm type. The algorithms broadly fall into two categories: supervised and unsupervised learning. Out of 51 software tools in this survey, 17 used supervised learning algorithms, and 34 employed unsupervised learning algorithms (Figure 2.6).

2.3.2.1 Supervised learning methods

Supervised learning methods aim to solve classification and regression problems. These algorithms require training data with known outcomes to learn from, in order to build a model to classify new inputs. In practical FC applications, manually annotated cell populations associated with healthy or diseased patients could be used as training data. Cell marker expression features that correlate with the two outcomes would be extracted from the data and then a model built to classify the disease status of new samples.

The limitation of these methods is that the algorithm is only as good as the training datasets available for it to learn from, and it is also possible to overtrain a learning algorithm. Furthermore, there are insufficient publicly available training datasets for all possible scenarios in clinical settings, especially those focussed on rare cell identification. The FlowCAP-II sample classification challenge used three real-world patient datasets, half of each dataset (training set) was labelled with patient clinical outcomes and the challenge was to correctly label the other half (test set). The comparison study found many algorithms achieved perfect classification accuracy on two datasets (acute myeloid leukaemia detection and HIV vaccination antigen stimulation groups), but all performed poorly on a third (HIV exposure on African infants) [49]. Because the current number of supervised learning software in FC data analysis is low, and there is limited availability of large training datasets, the majority of this analysis concentrates on the significant number of unsupervised methods.

2.3.2.2 Unsupervised learning methods

With unsupervised learning, no training dataset is needed, and the goal is to correctly identify and quantify cell populations in FC data. Automated gating of cell subtypes is viewed as a clustering problem. The unsupervised learning software tools in this survey apply different clustering methods such as hierarchical clustering, partition clustering, model-based clustering, density-based clustering (Figure 2.7). Dimensionality reduction is also used to simplify multiparameter datasets. Below is a brief overview of the most fre-

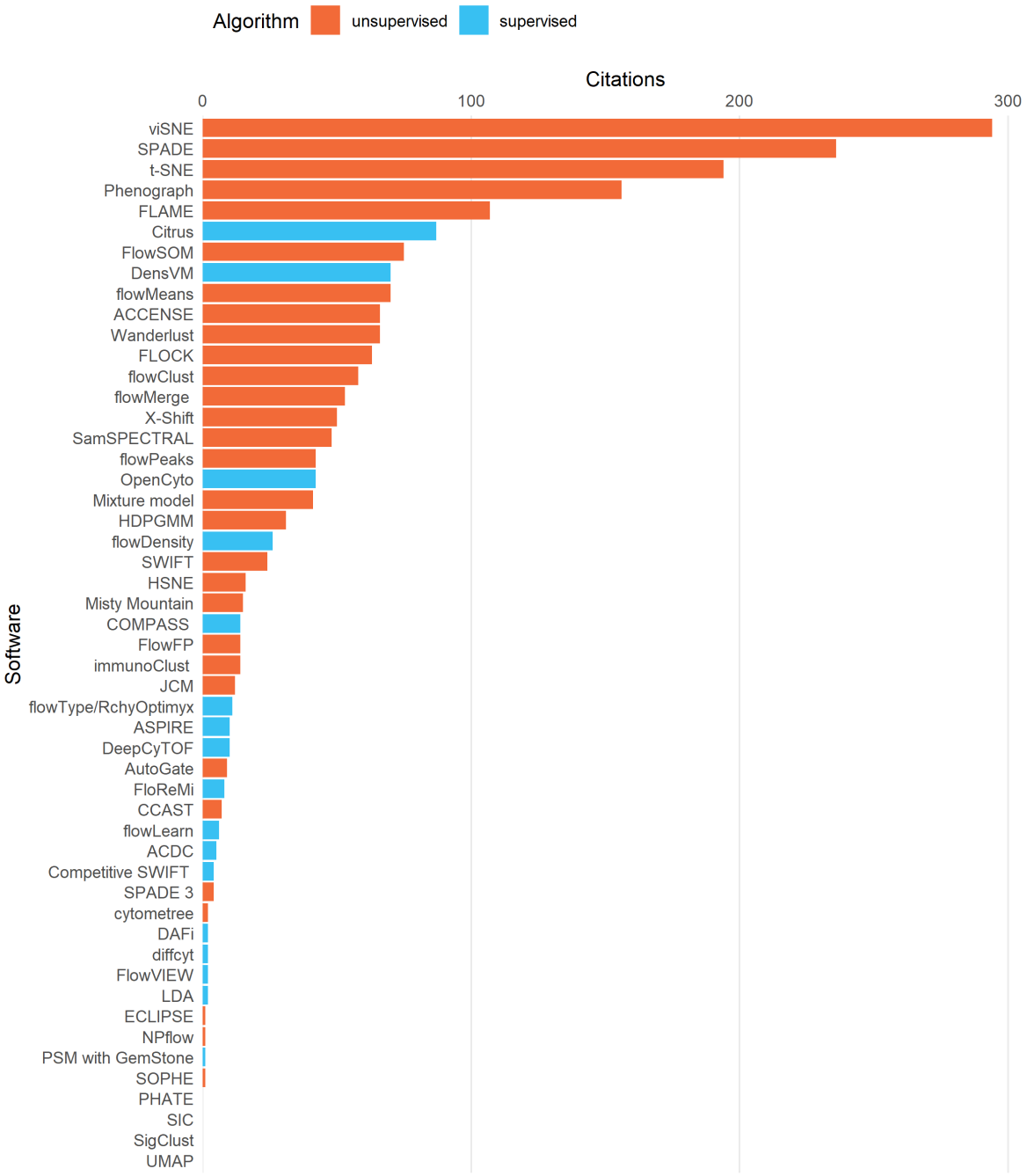


Figure 2.6: Number of citations by machine learning method. Unsupervised learning methods include clustering, dimensionality reduction and do not require training data. Supervised learning methods include classification and regression such as support vector machines, artificial neural networks. They require manually labelled training data to build a model and preform predictions. The most frequently cited flow cytometry software algorithms apply unsupervised learning approaches.

quently used clustering algorithms. For a comprehensive survey of clustering algorithms, see [120].

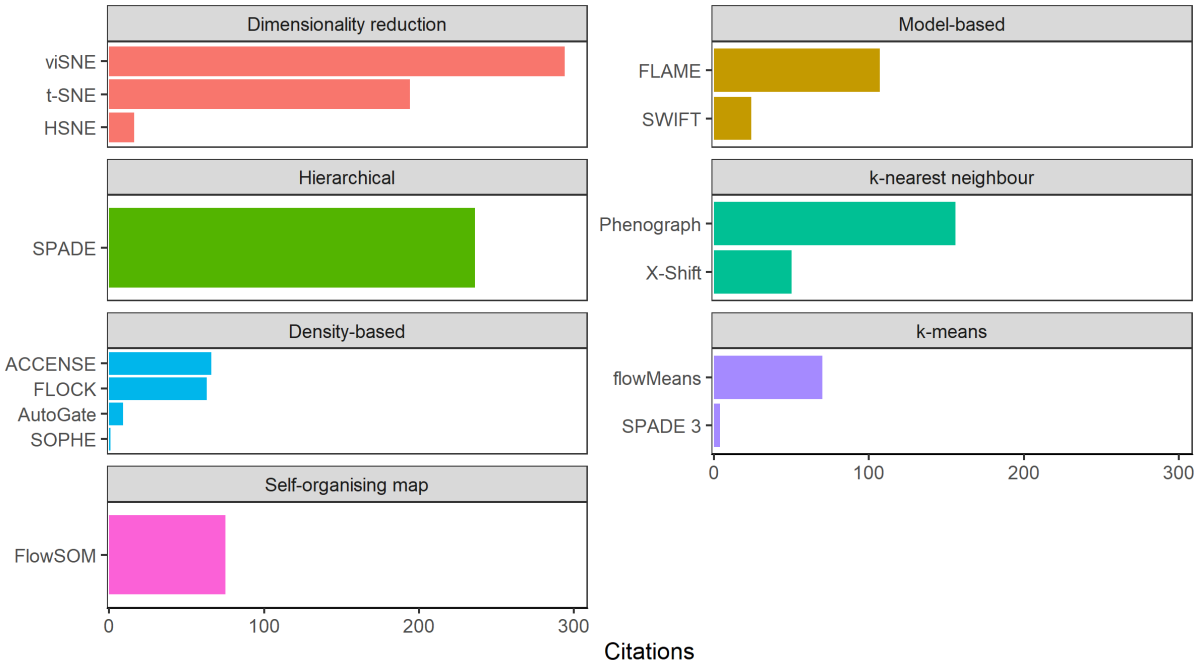


Figure 2.7: Software citations by computational method (unsupervised methods with GUIs only).

Hierarchical clustering Hierarchical clustering has two strategies to group similar datapoints together, agglomerative and divisive [121]. The agglomerative method follows a bottom-up approach, where neighbouring datapoints are merged to form sequentially larger clusters, until only one cluster remains. The divisive method follows a top-down approach, starting with the whole dataset as one cluster and partitioning it to form smaller clusters down to the level of individual datapoints. The target number of clusters is determined by the user. The resulting clustered data can be visualised as a hierarchical tree structure (dendrogram) which resembles phylogenetic trees. Thus, hierarchical clustering appears well suited to classifying datasets with evolutionary observations and may have natural uses for analysing cell development, maturation and differentiation data from time course experiments.

The second most frequently cited software tool in this survey, SPADE1, applies agglomerative hierarchical clustering in its algorithm [68]. A prior density-based down-sampling step is performed to equalise low density populations with high density ones. Down-sampling reduces the time complexity of the hierarchical clustering step, and also increases the prevalence of rare cell types and noise events. The SPADE1 algorithm

overcomes the problem of selecting the number of clusters by over-clustering the dataset (e.g. instead of 3 nodes, set 100 nodes). The algorithm builds a minimum spanning tree (MST) from the clustered data, and then relies on expert operator manual analysis to partition the MST to determine correct number of cell populations. An improvement on the SPADE1 algorithm, SPADE3, has been released to remove the stochastic nature of the original agglomerative algorithm by implementing a deterministic k-means clustering algorithm, and to introduce a semiautomated interpretation of the MST [105], thus creating a new software tool (albeit with the same name) with different mathematical definitions and characteristics, and potentially different data analysis outcomes. In addition to these algorithmic differences between the versions, SPADE3 is primarily implemented in Matlab although stand-alone executable code does exist, SPADE1 and its updated version SPADE2 (better GUI and runtimes) are implemented in R and are available on Cytobank and as a plugin on FlowJo.

***k*-means clustering** The k-means clustering method was first published in 1955 and is one of the most popular clustering algorithms used in pattern recognition [122]. k denotes the number of clusters, which is user defined. The k-means algorithm begins with k seed points randomly scattered in the dataset acting as cluster centres. Neighbouring datapoints are assigned to their nearest seed to form the initial clusters. The centre of the clusters, the centroid, is calculated and repositioned. The algorithm repeats the assignment of datapoints to the updated centroid, and then updates the centroid, and so on. Further iterations to update the clustering are performed until cluster membership stabilises. k-means is an efficient algorithm, with faster run times compared with hierarchical and model-based clustering. However, the drawbacks are its requirement for a predefined number of clusters, its limitation to spherically shaped data and sensitivity to outliers. These are key issues that need to be addressed for correct analysis of FC data, which are usually non-convex shaped and noisy.

The software tool flowMeans [75] and flowPeaks [83] are based on k-means clustering, and attempt to solve these limitations of k-means clustering on FC data by over-clustering the data then merging nearby clusters to obtain a single population. flowMeans applies a change point detection algorithm to detect the number of clusters, whereas flowPeaks fits a Gaussian finite mixture model to the initial k-means clustered data then generates a density function to search and merge peaks. The results successfully identify non-spherical cluster shapes, however, rare clusters remain difficult to uncover.

***k*-medoids clustering** k-medoids clustering, also known as partition around medoids (PAM), is similar to the k-means method, intending to partition the dataset into k clusters, but instead of using centroids (the mean of the datapoints in a cluster) to assign nearby objects, k-medoids uses the representative object of a cluster with minimal average dissimilarity to its assigned objects [121]. k-medoids is less sensitive to outliers than k-means, however, its main disadvantage is the high computational cost for analysing large datasets. Sampling of the dataset is one strategy to reduce runtimes (CLARA) [121]. A modified version of PAM has been proposed for use in a clustering analysis pipeline to identify cell populations [117].

***k*-nearest neighbour (*k*NN) graph** This method, not to be confused with k-means, defines k as the number of nearest neighbours a single datapoint has (using a distance metric, such as Euclidean distance). The datapoints (nodes) are connected by edges to build a graph. For example, setting a k of 5 connects each datapoint with its five nearest neighbours. When performed for the whole cytometry dataset, this results in dense areas appearing where cells are more phenotypically similar to each other. The key issue with this method is selecting an appropriate value for k. PhenoGraph [70] applies the k-nearest neighbour method to construct a graph, then partitions the graph into communities using the Louvain community detection method [123] in order to identify distinct subpopulations. X-shift applies the kNN density estimation method to cluster data [81].

Density-based clustering Density-based clustering algorithms such as density-based spatial clustering of applications with noise (DBSCAN) [124] and ordering points to identify the clustering structure (OPTICS) [125] views datapoints in high density regions as clusters, separated by regions of low density. Density-based clustering identifies core points belonging to a cluster as well as noise points. These algorithms are intended to discover clusters of arbitrary shape, such as geographical data. Key requirements are a threshold for the minimum number of points in a neighbourhood and an arbitrary distance measure for the density-reachability of a point to a core point. Since the number of clusters is not a required input parameter, this method is useful for FC data analysis where the number of cell subtypes is unknown. Generically, density-based clustering algorithms appear to be a widespread strategy for software developers to identify cell populations, and are used by several software tools: ACCENSE [76], DensVM [74], Flock [78], flowDensity [87] and Misty Mountain [91], noting that mathematical implementations and algorithms may vary depending upon the data analysis approach.

Model-based clustering Model-based clustering assumes the data follows a statistical distribution and models this onto the dataset. For example, Gaussian mixture modelling (GMM) views the data as consisting of several Gaussian (normal) distributions and merges the data to the predetermined number of clusters fitting the model. There are numerous mathematical models available, so basic problems arise in selecting an appropriate model and choosing the number of clusters for fitting the model. The optimal model neither underfits nor overfits data, and can be estimated using criteria such as the Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC) [126]. This approach to fit each model to the data to find the best fit is computationally expensive.

Model based clustering methods are the most frequent in this survey and may be due to the plethora of statistical models to choose from. These include models based on mixtures of Gaussian distributions, Student's t-distributions and skew t-distributions. The following examples of software tools use model-based clustering methods: FLAME [71], flowClust [79], flowMerge [80], SWIFT [88, 89].

Spectral clustering Spectral clustering is based on graph theory where each datapoint represents a node, and the edges are weighted based on a similarity criterion. Clustering is achieved through graph partitioning [127]. Spectral clustering is used by the software tool SamSPECTRAL [82] which includes a subsampling step to reduce runtimes.

Self-organising map The self-organising map (SOM) is based on a model of neural network learning [128]. The premise is to construct a grid and map random datapoints one at a time onto each node of the grid. The grid self-organises so that neighbouring nodes have greater similarity, and less similar nodes are moved further away. The next input datapoint is applied to the node that matches best with it. In the end, a large high dimensional dataset is reduced to a low dimensional space while retaining the global structure of the original data [129]. The resulting SOM can be clustered further to group similar nodes, using traditional methods such as hierarchical agglomerative clustering and k-means clustering [130]. The FC data analysis software tool FlowSOM builds a minimal spanning tree from the SOM, followed by a consensus hierarchical clustering step to give the expected number of cell types [73].

Dimensionality reduction Dimensionality reduction is not strictly a clustering method. The idea is to take data containing multiple parameters and reduce it to (usually) two dimensions which can be easily interpreted. Principle component analysis (PCA) is an established dimensionality reduction method, however, newer algorithms such as t-

stochastic neighbourhood embedding (t-SNE) are a significant improvement that preserves (to a limited extent) both the local and global structure of the high-dimensional data, and generates a visual map of the data where similar points are clustered together [69]. Albeit very large datasets ($>10^6$ events) can cause crowding in the layouts that limit meaningful interpretation of the data, and runtimes are slow [118]. The t-SNE algorithm and its implementation in viSNE successfully visualises a variety of large real-world datasets and appear well suited to analysis of large multidimensional FC data [67]. This is reflected in their overwhelming popularity in this survey with viSNE and t-SNE ranking first and third respectively in the software citation analysis, and their numbers combined make up 24% (488 out of 2,072) of all citations. Dimensionality reduction is increasingly being used as the first step of a data analysis pipeline to extract initial clusters, followed by a clustering step to identify cell populations [90].

The benefits of data visualisation and interpretation following dimensionality reduction have encouraged further development of similar data analysis tools that improve scalability, runtimes and are better able to handle large ($>10^6$) datasets and represent the global structure. These tools include hierarchical stochastic neighbour embedding (HSNE) [131], PHATE [115] and uniform manifold approximation projection (UMAP) [118]

2.3.2.3 Pre-processing tools

Although excluded from this study, automated pre-processing tools play an important role in FC data analysis because they enable high-quality input data for all the analysis approaches mentioned above. Pre-processing tools used to clean raw data include quality control tools to remove fluorescence anomalies (flowClean, flowAI), perform transformation (flowCore) and normalisation (flowStats) [132, 133, 134, 135]. Manual gates that exclude doublets, debris and dead cells can be imported from FlowJo into R using flowWorkspace [136], and these manual gates can also be automatically replicated using flowDensity [87].

2.3.3 Updates to literature survey

2.3.3.1 New software tools

Additional software tools have been released since the literature survey was performed at the early stage of this research project in 2019. A search for new and noteworthy software tools returned the following examples:

- Several optimised implementations of t-SNE: FIt-SNE (Fast Fourier Transform-accelerated Interpolation-based t-SNE) [137], opt-SNE [138], and qSNE [139].
- Infinity Flow, a supervised machine learning tool intended for analysis of expression levels of hundreds of proteins across millions of single cells [140].
- A 'data fusion' method based on discriminant analysis to classify cell subsets and predict immune responses [141].
- FAUST (Full Annotation Using Shaped-constrained Trees), a machine learning method implementing algorithms for clustering, cluster matching, variable selection, and feature selection [142].
- AutoSpill, a method for automatic compensation in data pre-processing [143].

2.3.3.2 Citations

Software tools that received very few citations at the time have accumulated more citations since, as the field evolved and uptake among users for single-cell analysis increased. In particular, the dimensionality reduction tool UMAP [118] has rapidly amassed 186 relevant 'cytometry' article citations by the end of 2021 — up from having no citations when it was newly published in 2019. A full update on the number of citations for all software tools was not performed because of time limitations in this research project. Nevertheless, the major findings and the general landscape of software tools have not changed; dimensionality reduction tools remain the most used, and unsupervised learning tools remain more widely used compared to supervised learning tools.

2.3.4 Summary of literature search on automated software tools

In summary, the popularity of FC automated data analysis software tools may depend on the convenience of having a GUI. Currently, unsupervised learning methods receive more citations than supervised methods. Among unsupervised methods, dimensionality reduction algorithms are more popular than other clustering algorithms, because it seems users value the automatic visual output of high-dimensional data presented in an intuitive way that retains local and global structure. Among the other unsupervised methods there was no specific class of algorithm that was more popular than others, although analysis methods that provide novel data visualisations (e.g. SPADE1, Phenograph, FlowSOM) received more citations than algorithms in the same class. A caveat in focussing on the popularity of a tool is that it does not necessarily provide information on its fitness for purpose, in this regard further investigations on performance is the subject of Chapters 4 to 7 of this thesis.

2.4 Clinical laboratory users survey

To obtain a full picture of the popularity of automated flow cytometry data analysis software tools, it was important to gain insight on their actual use within clinical centres, not apparent from literature citations. An invitation to participate in a survey was distributed to laboratories worldwide registered with the EQA/ proficiency testing programme from UK NEQAS for Leucocyte Immunophenotyping [144]. The survey aimed for a broad overview and was not intended to extract actual participant use of specific functions of software tools. Survey distribution occurred in January 2020 and responses were gathered over 1 month. The online survey of 8 questions (Table 2.2) was developed to expand on the literature review to understand the potential use of automated software tools in clinical laboratories.

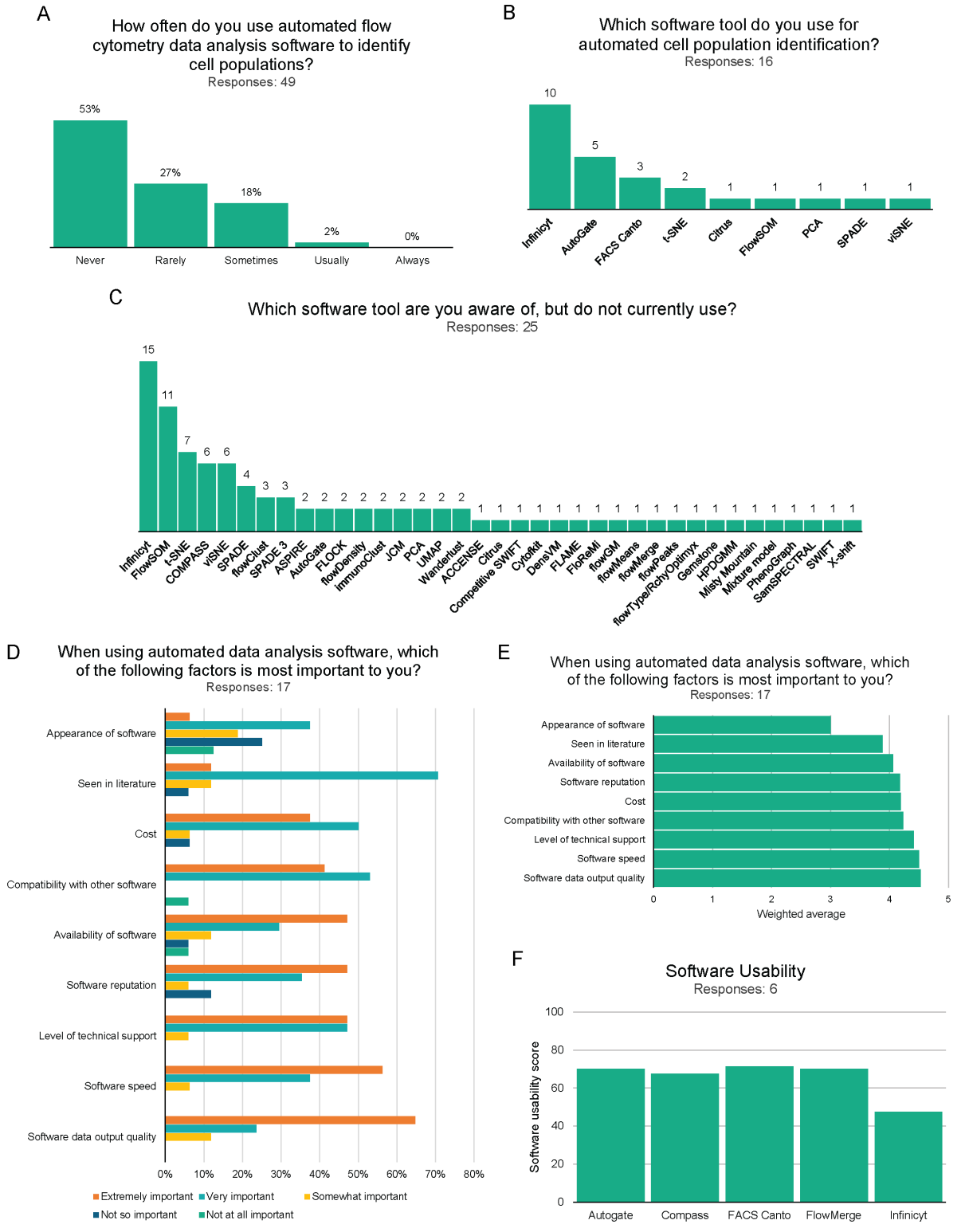


Figure 2.8: Results of a survey of clinical laboratories on the use of automated flow cytometry software tools.

Table 2.2: Survey questions and answer response choices.

Q1	In a typical week, how many hours do you spend analysing (gating) flow cytometry data on a computer? • Over 30 hours • 20-30 hours • 10-20 hours • 1-10 hours • Less than 1 hour
Q2	How often do you use automated flow cytometry data analysis software to identify cell populations? • Never – I only use manual gating to identify cell populations. • Rarely – I mainly use manual gating, but occasionally use automated tools. • Sometimes – I split my analysis equally between manual and automated cell population identification. • Usually – I mainly use automated tools, but occasionally use manual gating. • Always – I use automated tools for all my data analysis.
Q3	Which software do you use for manual cell population identification? (Check all that apply) ☐ BD FACS Diva ☐ BD FACS Canto ☐ BD FACSuite ☐ BD CellQuest ☐ FCS Express ☐ FlowJo ☐ FlowLogic ☐ Infinicyt ☐ Kaluza ☐ Navios ☐ VenturiOne ☐ WinList ☐ Other (please specify)

Table 2.2 – continued from previous page

Q4	Which software do you use for automated cell population identification? (Check all that apply) (Answer choices as in Table 2.1)
Q5	Which automated data analysis software are you aware of, but do not currently use? (Check all that apply) (Answer choices as in Table 2.1)
Q6	When using automated data analysis software, which of the following factors is most important to you? (Answer choices for each factor: Not at all important/ Not so important/ Somewhat important/ Very important/ Extremely important) • Appearance of software • Availability of software • Compatibility with other software • Level of technical support • Seen in literature • Software data output quality • Software reputation • Software speed • Cost • Other (please specify)
Q7	Please select the automated cell identification tool you are most familiar with. (Answer choices as in Table 2.1)

Table 2.2 – continued from previous page

Q8	Please mark your response about the software in Q7 to the following statements: (Answer choices for each statement: Strongly disagree/ Disagree/ Neither agree nor disagree/ Agree/ Strongly agree) a) I think that I would like to use this software frequently. b) I found the software unnecessarily complex. c) I thought the software was easy to use. d) I think that I would need the support of a technical person to be able to use this software. e) I found the various functions in this software were well integrated. f) I thought there was too much inconsistency in this software. g) I would imagine that most people would learn to use this software very quickly. h) I found the software very awkward to use. i) I felt very confident using the software. j) I needed to learn a lot of things before I could get going with this software.

2.4.1 Survey results

The survey received 49 responses out of 310 potential respondents, a response rate of 16% which is consistent with typical response rates of 15% to 20% from email invitations to participate in online, non-incentivised surveys [145]. The quality of respondents is high because of the targeted nature of the survey to subscribers of an EQA programme. Although conclusions from 49 responders should be considered with care, the survey is valuable in providing strong suggestions of behaviour on the current use of automated FC tools in clinical laboratories. The survey found more than half of respondents (26 out of 49, 53%) never use automated FC software tools and only use manual gating to identify cell populations (Figure 2.8A). 13 out of 49 (27%) mainly use manual gating but occasionally use automated software tools, and 9 out of 49 (18%) split their analysis between manual and automated methods. One respondent mainly uses automated software tools but occasionally use manual gating.

The results suggest (on this basis) that clinical laboratories often rely on manual gating to identify cell populations and the use of automated methods have yet to be firmly

established. The observed pattern of adoption is expected given the emerging nature of the software tools. The survey asked participants to identify which automated data analysis software tools they used (Figure 2.8B). Nine software platforms were identified among the 16 respondents who used automated software tools, the most frequently identified of which was Infinicyt (63%). Other software tools selected included AutoGate (31%) and FACSCanto (19%).

The survey also asked participants to identify software tools they were aware of but do not currently use (Figure 2.8C). A total of 37 automated data analysis tools were identified by respondents, an increase of 28 from the number of software tools respondents actually used. Once again, Infinicyt software was the most popular response (60%), followed by FlowSOM (44%), t-SNE (28%), viSNE (24%), and COMPASS (24%). For further insight, responses were grouped according to manual-only users (never use automated software tools) and automated users. This grouping revealed the automated-user base of respondents were aware of a wider range of software tools available compared with manual-only respondents: 36 software tools were identified by (13 out of 23) automated users compared with 8 identified by (11 out of 26) manual users.

The results gathered from this question suggest many laboratories were aware of what software tools were available but perhaps have not had the time or resources available to validate and implement changes to a manual gating protocol to incorporate automated analysis. It is also possible that laboratories first consider the many software packages available before committing to purchase only one software package, such as Infinicyt. Furthermore, software tool selection may be partly influenced by common consortium recommendations or EQA schemes.

To understand the factors which users consider important when using automated software tools, survey participants were asked to grade the importance of factors along a 5-point scale from 'not important at all' to 'extremely important' (Figure 2.8D, E). Results from this question revealed the most important factors for users was the software data output quality, followed by software speed, and the level of technical support. Of lesser importance, scored in decreasing order, were factors such as compatibility with other software, cost, software reputation, software availability, and, seen in the literature. The appearance of software was the lowest ranked importance factor in the survey.

To further understand the user interaction with automated software tools and the potential impact this has on software selection, development and data quality, the survey participants were asked in Question 8 to assess the software tools they were most familiar with by responding to 10 usability statements on a 1 to 5 score scale from 'strongly disagree' to 'strongly agree'. The statements are based on the System Usability Scale (SUS)

and were designed to provoke extreme disagreement or agreement among all respondents [146]. Statements that commonly lead to strong disagreement alternate with those that lead to strong agreement, to prevent response biases. This arrangement allows calculation of the SUS score, where a) the score of each odd-numbered statement minus 1, and b) the score of each even-numbered statement taken away from 5, are summed then multiplied by 2.5 to obtain a score out of 100, with higher scores indicating better usability. Scores for individual statements are not meaningful on their own and need to be taken together to give a measure of the overall software usability. On the basis of an extensive literature search that found no results, this appears to be the first application of the SUS to quantify the usability performance of flow cytometry automated software tools.

This question received 6 responses ranking 5 software tools (Figure 2.8F). From the individual surveys, AutoGate, FACS Canto and FlowMerge received SUS scores above 70, therefore were judged to have 'acceptable' usability based on the benchmark provided by Bangor et al. [147]. Compass received a SUS score below 70, indicating 'marginally acceptable' usability. Infinicyt received a SUS score below 50, falling into the 'unacceptable' region. Whilst the number of responses to this question were too low to draw conclusions from, it was interesting to note that the most identified software tool among the survey was also the least user friendly, and as the field of computational cytometry is anticipated to mature and user uptake to increase, these initial SUS scores calculated will provide a critical baseline for future benchmarking studies to compare against.

The clinical survey results showed that 16 respondents identified 9 software tools they make use of, and 25 respondents identified 37 software tools they were aware of but do not use. Excluding duplicates, 38 unique software tools were identified. A cross comparison with the 51 software tools identified from the literature review reveal the majority (36 out of the 38) were included in both surveys. Two software tools do not appear in the literature review, which happened to be commercially available ones. This shows good correlation between the two information streams. The analysis of literature captured the software tools used at a point in time (over 10 years) that precedes current usages, whereas the clinical survey revealed the most up to date patterns of use. Because of this contrast in timepoints, the clinical survey captured only 2 more software tools that slipped through the literature review.

The most frequently identified software tool by the online survey, Infinicyt, was not identified as a function of the original literature search strategy and not mentioned in previous reviews on automated analysis tools. Infinicyt is proprietary software for analysis of multidimensional flow cytometry data, developed with support from the EuroFlow Consortium for standardisation of immunophenotyping protocols [53]. The main feature

of Infinicyt is the supervised learning algorithm for automatic identification and classification of cell populations based on reference databases built from merged multicentre patient files [148, 149]. Application of these Infinicyt tools are optimised to samples acquired following fully standardised EuroFlow standard operating procedures, reagents, instrument settings and 8-colour antibody panels for haematological malignancies [150]. The database-guided tool has been shown to successfully classify acute leukaemia cases using a database constructed from 656 patients [149]. The software is also designed to be integrated with a laboratory information system (LIS) for secure handling of patient data. The highly specialised purpose of Infinicyt for clinical diagnostics explains its common use in clinical laboratories survey, and perhaps its under representation in research areas.

Another popular software tool among the clinical laboratories, FACSCanto, was not featured in the original literature search because of the lack of peer-reviewed published work on its automated cell population identification function, but the clinical survey has identified it. The survey participants used FACSCanto software for analysis of CE-in vitro diagnostic (IVD)-marked assays such as CD4 and CD34 absolute count analysis. The software tool provides automated analysis of workflows and, similar to Infinicyt, is designed for clinical cytometry with LIS enabled connectivity. FACSCanto software popularity is possibly influenced by its bundled distribution with BD cytometer equipment and is therefore used by default by operators.

Common software tools highly ranked in both the literature citation analysis and the online survey were: FlowSOM, t-SNE, viSNE, and SPADE1. Overall, although uptake of automated software tools is growing, manual gating remains the standard practice. For clinical laboratory users, the most important component of automated software tools is the data output quality. This factor was not obvious from the findings from the literature citations. For automated analysis techniques to overtake manual gating, not only do the cell population identification results have to replicate expert manual analysis, but the results obtained from algorithms must also be robust with cell population numbers that can be reported with confidence.

2.5 Manufacturing survey

An attempt was made to identify the most popular software tools in biomanufacturing spaces, with survey distribution occurring through the UK Cell and Gene Therapy Catapult network. This exercise resulted in one response, which was insufficient to draw any conclusions from, and so this arm of the study was discontinued. Possible reasons for the low response rate may be that the survey was not widely distributed enough or not tar-

Table 2.3: Selected software tools for thesis

Software tool	**Clustering method**
FLOCK	Density-based
flowMeans	*k*-means
FlowSOM	Self-organising map
PhenoGraph	*k*-nearest neighbour
SPADE1	Hierarchical
SPADE3	*k*-means
SWIFT	Model-based

geted to the relevant personnel, alternatively the survey was received but not completed because potential participants were too busy, lacked incentives, or had reasons not to participate in the research e.g. to protect commercial interests.

2.6 Software tool selection

In the context of this thesis, this literature review enabled the shortlisting of software tools to take forward into comparison studies. Unsupervised learning tools were selected rather than supervised learning tools, because this review identified greater maturity in the development of the former. Dimensionality reduction tools, although popular among users, were excluded because their mathematical differences to clustering algorithms would require significantly different research testing methodologies.

The most highly cited software tools from each class of clustering algorithm was selected (see Figure 2.7). Software tools that were unable to be run successfully were rejected, and the next most cited tool in its group was selected instead (FLAME replaced by SWIFT for model-based clustering; ACCENSE replaced by FLOCK for density-based clustering). In addition, two versions of SPADE were included to explore potential variability between them. The shortlisted software tools are listed in Table 2.3, alongside their clustering algorithm employed.

2.7 Discussion

Flow cytometry has evolved to a stage where data analysis can be approached with unsupervised and supervised learning methods that automatically cluster cell populations and classify samples corresponding to clinical outcomes. Automated techniques allow FC analysis without manual variability, subjectivity, and bias of gating, and thus many new methods have been developed in the field in the past decade. However, it should be recognised that many of the automated techniques require moderate to significant operator control of software variables (beyond the default settings) and hence human subjectivity within the data processing chain may still be apparent.

In this literature survey, the current state-of-the-art software tools have been identified and their popularity ranked based on literature citations. Although citation counts do not necessarily reflect the use of software tools in labs, they give a good indication. The purpose of this study was to define the prevalence and perceived volume of use of automated software tools, not specifics of use in a laboratory or manufacturing company. Highly ranked software tools included: viSNE, t-SNE, SPADE1, PhenoGraph, FlowSOM and Citrus. A common attribute of these software packages is the availability of a GUI that increase ease of use and appearance. This highlights the importance of usability as a factor for uptake of automated software tools in the community. Moreover, these software tools are implemented in multiple platforms (Bioconductor, FlowJo, Cytobank), and provide novel visualisation outputs to aid interpretation of the data. Trends between software frequency of citation and factors such as cost or the underlying algorithm type were not apparent.

In addition to the literature survey, an online questionnaire of clinical laboratories on the use of automated FC software tools was completed via the external quality assessment (EQA)/ proficiency testing programme from UK NEQAS for Leucocyte Immunophenotyping. This survey collected actual real-world usage data and opinions about automated FC data analysis software tools from a global targeted audience which could not be obtained from the literature search. Noting that this analysis was based on 49 respondents out of a possible 310 participants, a strength of this survey lies in its distribution through the EQA network rather than a public medium, which was more likely to ensure genuine trustworthy responses. Very few surveys of this nature have been published in the literature.

The online questionnaire did not capture users in similarly highly regulated spaces such as biotechnology, pharmaceutical and contract research organisations. However, distribution of a survey to those parties will be more difficult because they do not neces-

sarily subscribe to a comparable EQA network, so networks from the International Society for Advancement of Cytometry (ISAC) and the International Clinical Cytometry Society (ICCS) could potentially be explored in the future. Most frequently identified automated software tools for clinical cytometrists were Infinicyt and FACSCanto, noting that 53% of participants stated that they never used these automated tools. Infinicyt in particular makes use of large reference patient databases to classify new patient samples using a supervised learning algorithm. These software tools have highly specialised workflows for analysis of regulated clinical assays to automated immunophenotyping, along with an important feature to connect with a hospital laboratory information system (LIS) to securely manage patient data.

The contrast in software tool popularity between the two complimentary surveys reflect the different needs and behaviours of the two communities. Clinical users are more likely to run routine, well defined assays with standardised processes to enable confident diagnostics of patient samples. For example, the highly standardised ISHAGE protocol for enumeration of haematopoietic stem cells in peripheral blood recommends the use of specific antibody conjugates and prescribes manual gating strategies to identify target cell populations [39]. In this respect, clinical users lean towards tools that replicate expert manual gating and can automate targeted analysis of well-defined populations. This is different in academia, where research is performed on well-defined cell subsets alongside unknown target cell populations, and hence users make more use of automated tools that support discovery and exploratory research.

The standardised datasets produced across clinical settings with the same experimental parameters, and crucially linked with specific patient outcomes, can be grouped to build a large database collection that allows for their use as training datasets for the development of supervised learning algorithms. In comparison, the academic space is less likely to have a large and diverse resource of labelled data to use for training purposes, and therefore is dominated by use of unsupervised learning methods. Overall, there is no 'best' method. The most suitable automated analysis tools to use will be context dependent, on factors such as cell type, the data structure and the purpose of the analysis. The best case is to provide users with complete details of how tools work, for them to make a well-informed decision. This may call for additional benchmarking methods/results from a wider selection of datasets.

More than half of the respondents from the clinical survey never use automated analysis tools and only use manual gating protocols, suggesting barriers to adoption of software tools may be widespread. The questionnaire gave an insight into the clinical users' software tool preferences when incorporating automated workflows into their data analysis.

High value was given to the data output quality, speed of software and level of technical support. The low take-up in automated software tools may be down to shortcomings in all three factors in the current options available. The most critical factor, quality of the data, is a major driver for the use of automated software tools. Tools that aid rigour and reproducibility are expected to be welcomed, so it is intriguing that adoption rates are low, but it may be down to human sentiment and trust in manual methods.

With respect to the speed of software tools, because results need to be reported in a timely (or possibly urgent) manner for clinicians to make decisions on patient treatment strategies, the analysis time needs to be in the order of seconds and minutes rather than hours and days. Current automated software tools may not offer significantly faster gains in analysis times over manual analysis that would incentivise uptake. Finally, better documentation in the form of detailed user manuals, video tutorials and troubleshooting guides would increase the level of technical support, and make automated analysis more widely used.

Regulatory requirements are a possible factor for the low uptake of automated methods in the clinical laboratory. Implementation of new diagnostic methods is driven by international guidelines (e.g. World Health Organisation (WHO), International Council for Standardization in Hematology (ICSH), International Clinical Cytometry Society (ICCS)). Consensus guidelines regarding the use of automated methods have yet to be established. Even once guidelines are published, implementing new protocols at the laboratory level requires documenting process change controls, validations, and verifications in line with quality management system ISO 15189:2012 [151]. The increased regulatory requirements in clinical spaces compared with academia may be a barrier to uptake. Diagnostic methods are typically developed on an individual disease or biomarker basis, so are narrow in scope by nature. This means the pace of automated adoption occurs one test at a time, rather than all the tests involving flow cytometry changing to automated analysis at once.

As the burden of manual analysis increases with the number of parameters in a panel, perhaps clinical laboratories with more complex panels will be keener adopters of automated software tools that offer more efficient, scalable and unbiased analyses. The awareness of new tools can be more dated among the clinical workforce because day-to-day sample processing demands reduces the time available to keep up to date with the latest literature. There are now trends for academic users to acquire programming skills in R, Python and Matlab to keep up with data analysis requirements. This is a less likely scenario in clinical laboratories and may be the reason for the lower uptake of tools that are executed in those programming environments.

To a certain degree, usage of these tools relies on the efforts of commonly used standalone software packages (e.g. FlowJo, FCSExpress) to implement automated tools as plugins integrated into their GUIs. The skills shortage presents a risk to employers, whether to train up staff to be knowledgeable in coding but lose that tacit knowledge when they leave the company, or to buy in a ready-made software tools with full GUI that does not require specialist training and is easy to learn for new users. Indeed, this study has shown a user preference for tools with GUI. The implication could be for high performing software tools without a GUI losing ground to lower quality but easier to use software tools.

This review has investigated the current usage trends and popularity of automated flow cytometry data analysis software tools. However, it is worth emphasising that the popularity of a tool does not indicate whether it is the correct or best approach of analysing data, and therefore a key question that has emerged from this study is whether popularity translates to quality. It is clear that challenges in the data output quality from automated software tools remain a hurdle to the widespread uptake of software tools in flow cytometry. This is an opportunity for this research to assess the actual performance of different algorithm types through a range of benchmarking real-world experimental and simulated datasets with controlled cell characteristics.

2.8 Chapter conclusions

- A comprehensive overview of currently available automated flow cytometry data analysis software tools was carried out, with over 50 tools being identified based on a literature search.
- The five software tools with the highest citation rates were: viSNE, PhenoGraph, SPADE, t-SNE and FlowSOM.
- Software tools analysed based on presence of GUIs, cost, and implementation of supervised or unsupervised learning algorithms.
- Unsupervised learning-based tools were further grouped into different mathematical approaches of: density-based clustering, dimensionality reduction, hierarchical clustering, k-means clustering, k-nearest neighbour, model-based clustering, and self-organising map.
- A survey distributed among clinical laboratories found 53% of users never use automated tools and only use manual gating to identify cell populations.
- The main challenges faced by users on application of automated tools include (in decreasing importance) concerns on the data quality, speed of software, and level of

technical support.

- Based on this work, seven representative automated tools were selected for further studies going forward in this thesis (Table 2.3).

Chapter 3

Synthetic datasets

The publication listed below was an outcome of the work reported in this Chapter:

> **Cheung M**, Campbell JJ, Thomas RJ, Braybrook J, Petzing J. Systematic design, generation, and application of synthetic datasets for flow cytometry. PDA Journal of Pharmaceutical Science and Technology. 2022. https://doi.org/10.5731/pdajpst.2021.012659

3.1 Introduction

Synthetic datasets are datasets generated by computer simulation, rather than collected through real world observations or experiments. These datasets are often created from mathematical models that approximate aspects of real-world data. Synthetic datasets can be referred to as 'simulated', 'artificial', 'mock', 'toy', and more colloquially, 'dummy data'. In unsupervised machine learning, well known 'toy' datasets used to compare different clustering algorithms include clusters in the shape of two rings, crescent moons, spirals, and data with no structures [152]. Sophisticated synthetic datasets include urban street images applied to object detection for autonomous driving [153, 154], and household objects images for object detection in the field of robotic manipulation [155]. In medical imaging fields, synthetic datasets generated based on real images (e.g. magnetic resonance imaging (MRI), mammography, and whole-slide histopathology datasets) have demonstrated utility within computer-aided detection or computer-aided diagnosis systems, and may be useful for educational purposes and in quality control [156, 157, 158].

Further strategies to generate artificial datasets have made use of data augmentation methods [159]. Similarly, in flow cytometry, cell subsets from real samples can be selected and computationally mixed in a copy-and-paste strategy with other real or synthetic cell

populations, to create augmented and reprocessed 'semi-synthetic' datasets [160].

Real data are generally required during the development of computational analysis tools to provide means for training and validation, as well as potential decision-making reasons. An analysis tool implies any method that performs detection, recognition, identification, classification, tracking, prediction, or any other function that enables subsequent decision making. Real data also play an important role in benchmarking studies that evaluate the performance of these tools. However, real data have various limitations that necessitates the creation of synthetic datasets to overcome them.

Often, real datasets with predetermined criteria are difficult to collect because of limited availability at certain conditions and time periods. Synthetic data can be designed to mirror existing real data and further optimise the dataset by including rare cases and those at extreme conditions, thereby enhancing the realistic range of features or parameters. Additionally, a high level of control is potentially achievable with synthetic datasets, where designers can quickly change one factor at a time or build up layers of complexity through controlled addition of factors.

Once collected, a major shortcoming of real data is the laborious and time-consuming task of labelling observations with meaningful information (e.g. healthy vs abnormal) performed by experienced personnel. Synthetic datasets can be designed with the labels inherent in the data, side-stepping this task. The desired property of the synthetic data is also known and can be applied in performance assessment of analysis tools.

A drawback of large-scale real datasets is that they sometimes take a large amount of time to acquire (particularly true concerning collection of rare events or disease states). An equivalent large-scale, complex synthetic dataset may also require a large amount of computing time to generate, however this problem is negated as computers become faster.

Further benefits of synthetic data are: the potential lower costs associated with use of a modern computer rather than expensive technical equipment, reagents and raw materials; the reproducibility of computer code; and the absence of personal data which means that the processing of synthetic data does not have the same privacy concerns and legal or ethical compliance requirements as that of real data [161].

In flow cytometry, synthetic datasets usually aim to mimic the properties of real cell populations. The properties of these randomly generated datasets range from simple two-dimensional datasets with four clusters [91], to up to 30 populations in 35 dimensions [81]. The statistical distributions of synthetic clusters vary from normal (Gaussian), to non-normal generated from mixtures of several Gaussians, and skewed [88, 71, 83]. Simulated background noise also features [82, 77]. Prior synthetic work approaches, however, have not explored other possible characteristics specifically such as distance between clusters

(both standard and rare), which is modelled in real data through the comparison of median fluorescence intensities between a stained and an unstained population in terms of population widths or standard deviations, in order to estimate the relative brightness of a fluorophore [162, 163]. Moreover, in a somewhat fragmented space, there is reason to apply systematic design on existing properties (such as the skewness of clusters) to optimise the coverage of characteristics.

Evaluation of developers' own tools using internally generated synthetic datasets is inherently biased, therefore external and independent testing is a prerequisite for software credibility in the clinical and biomanufacturing communities. Benchmarking datasets are used in independent studies to compare software performance, however, existing studies performed have solely relied on experimental data toolsets and have not used synthetic datasets [49, 47]. This may be related to a limited amount of synthetic datasets available within public flow cytometry repositories for the community to use [164]. Software benchmarking studies hold similarities to other quality assurance methods such as proficiency testing according to ISO 13528:2005 [165], as an external and independent assessment of the accuracy of software results. When using real datasets as the test material, determination of the software performance is achieved through comparison of software results against an estimate of the true value. This value is assigned through a choice between 1) formulation, 2) cellular certified reference materials (of which very few exist for flow cytometry) [166], 3) manually gated analysis from one expert, 4) consensus manual analysis results from a group of experts, or 5) consensus values from participant results. Possible bias from the results of experts or participants reduces the robustness of the test. Synthetic datasets can be used adjacent to certified reference materials with potential benefit.

This research proposes the use of synthetic datasets for benchmarking unsupervised learning automated flow cytometry data analysis software of which there are a large array of options available to the data analyst [167]. This Chapter begins with defining a description of the data characteristics of flow cytometry data and then demonstrates two methods to generate highly controlled, systematically designed synthetic datasets with different degrees of separation between clusters, and different levels of skew. Finally, the use of synthetic datasets is illustrated using an exemplar software, SPADE3 [105], and results that allow robust calculations of performance metrics not possible with real cell data are presented.

3.1.1 Chapter aims

The work carried out in the previous Chapter identified several candidate automated flow cytometry data analysis tools for comparison studies. However, for effective and informative comparisons to take place, it is crucial to also have testing datasets that are fit for purpose. This Chapter focuses on the synthetic datasets that are utilised throughout this thesis to test software tools. The methods to generate the synthetic datasets are outlined, and justification for their application in comparison studies within this thesis is provided. In addition, evaluation metrics for software performance are considered.

The aims of this Chapter are to:

- Examine the properties of real flow cytometry data that can be modelled in synthetic data.
- Develop an approach to generate synthetic datasets with systematically designed properties that mimic the key characteristics of real data.
- Validate the synthetic datasets using real flow cytometry data.
- Apply the synthetic dataset to test the performance of an exemplar software, SPADE3.
- Explore the advantages and disadvantages of using synthetic datasets in place of real world data for comparison studies.

3.2 Materials and Methods

3.2.1 A note on data structure and terminology

Biologists, haematologists, immunologists, and other experienced users of flow cytometry will most likely 'see' flow cytometry data in the form of dot plots, or some other 2D graphical display. In this Chapter on synthetic dataset, it is useful to take a step back, and first understand how data are structured in Data Science terms. In the concept of tidy data, information is structured in a table, where rows contain the observations, columns contain variables, and each value corresponds to both an observation and a variable (Table 3.1) [168]. Flow cytometry data are stored in a standard file format FCS 3.1, with experimental data recorded in list mode, defined as a linear array of vectors, each vector corresponding to an event and vector components corresponding to types of measurements [51]. Flow cytometry data are compatible with the tidy data concept, where each row contains a cell event, and columns containing measurements such as

forward scatter, side scatter, etc (Figure 3.1). By understanding the basic structure of flow cytometry data in table form, the construction of synthetic datasets through simple arrangements of rows and columns can be appreciated and demystified.

Table 3.1: Data structure terminology

General term	Tidy data term	Flow cytometry term
Row	Observation	Cell / Event
Column	Variable	Marker / Parameter
Cell	Value	Expression value

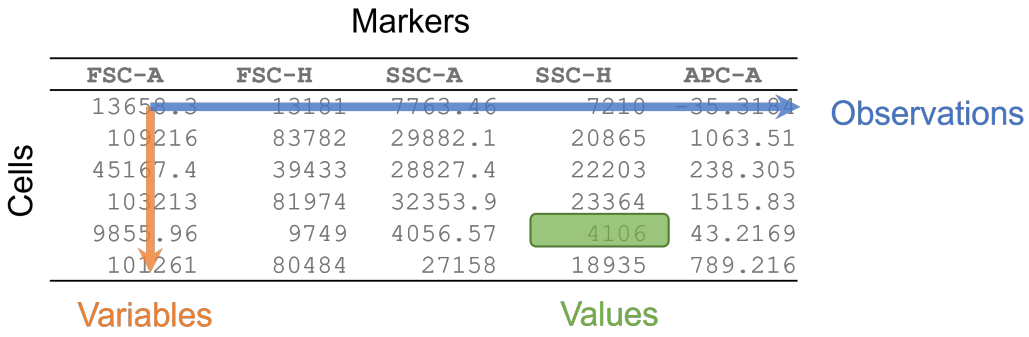


Figure 3.1: Flow cytometry data structure. Each cell event is represented along a row, these are the observations. Each marker is represented as a column, which form the variables. Each value corresponds to a cell and a marker. (FSC-A, forward scatter area; FSC-H, forward scatter height; SSC-A, side scatter area; SSC-H, side scatter height; APC-A, allophycocyanin area)

3.2.2 Target characteristics for synthetic flow cytometry datasets

Certain commonly recognised data characteristics or potential statistical attributes of flow cytometry data were identified and a strategy to control and modify these characteristics was put forward to create systematic scenarios for testing software (Table 3.2). In this research, the separation / overlap and the skew properties were targeted in simulation studies because these had not been addressed in previous work and/or the designs had not been approached in a systematic way. In order to focus on these properties, non-target characteristics such as the number of clusters, number of datapoints, and number of dimensions were kept constant, and noise was excluded in simulations (although this was subsequently investigated in Chapter 6).

Table 3.2: Characteristics of flow cytometry datasets

Characteristic	**Description**
Number of clusters	Number of cell subpopulations in a sample
Number of datapoints	Number of cell events acquired from a sample
Number of dimensions	Number of parameters recorded in the experiment, e.g. forward scatter, side scatter, fluorescent markers
Separation	The gap between negatively and positively stained cell populations
Overlap	Poorly resolved populations that appear merged
Placement	Projection direction of one cluster to another in space
Distribution	The shape of the cell population, as modelled on probability distributions e.g. Gaussian, Student's t, exponential, Chi-squared
Spread	The variance of the cell population
Skew and kurtosis	The level of asymmetry around the mean of the cell cluster
Orientation	The direction of the asymmetry
Elongation	Stretched out populations with long tails
Noise	Events that are excluded from analysis e.g. outliers, dead cells, debris, doublets, false events detected in the region of interest.

3.2.3 Hardware and software

Dataset generation and analysis was run on a 64-bit Windows 10 operating system with a 3.00 GHz processor and 64 GB of RAM. Computational tools used are listed in Table 3.3. Throughout this text, regular type is used to refer to software or computing environments, *italics* for packages, and `monospace` font to designate functions.

3.2.4 Synthetic datasets

The concept of creating artificial, computer-generated flow cytometry datasets is essentially random number generation, with numbers typically drawn from a normal distribution. Other probability distributions are available e.g. binomial, exponential, Poisson, Student's t, etc. If flow cytometry data are considered as mixtures of subpopulations of a heterogenous sample, then the generation of synthetic data is a process of creating a mixture of random clusters.

3.2.4.1 Single cluster simulation

A basic synthetic flow cytometry dataset can be simulated from a multivariate normal distribution. In statistics, a multivariate normal distribution is defined as a generalisation of a univariate normal distribution to higher dimensions [169]. A univariate normal (or Gaussian) distribution can be generated by defining the parameters of the mean of the distribution (μ), and its standard deviation (σ) (also its variance σ^2). This can be visualised on a probability density function plot as a characteristic bell curve (Figure 3.2).

For data containing two random variables X and Y (which can be plotted in 2D space to depict a cloud of points), the distribution can be described by the mean vector (μ), and the covariance matrix (Σ). The mean vector is comparable to the coordinates at the centre of the cloud:

$$\mu = \begin{bmatrix} \mu_X \\ \mu_Y \end{bmatrix} \tag{3.1}$$

The covariance matrix is symmetrical, with the diagonal elements specifying the variances of each variable, and the off-diagonal elements specifying the covariance between variables:

$$\Sigma = \begin{bmatrix} \sigma_X^2 & \sigma_X \sigma_Y \\ \sigma_Y \sigma_X & \sigma_Y^2 \end{bmatrix} \tag{3.2}$$

Table 3.3: Toolset used in this research for the generation of synthetic datasets, automated cell population identification, and performance evaluation.

Tool	**Version**	**Purpose in this research**
R	3.5.1	Programming
RStudio IDE	1.2	Programming environment
MATLAB	R2019a	Environment for SPADE analysis
FlowJo	10.6	Flow cytometry data analysis and visualisation
SPADE	3	Automated analysis of synthetic datasets
caret	6.0-82	Calculate performance metrics, confusion matrix
clusterGeneration	1.3.4	Generate synthetic clusters
flowCore	1.48.1	Manipulate flow cytometry data
psych	1.8.12	Measure skew
scales	1.1.0	Scale functions for visualisation
sn	1.5-3	Build and manipulate probability distributions of the skew-normal family
tidyverse	1.3.0	Data manipulation, analysis and visualisation

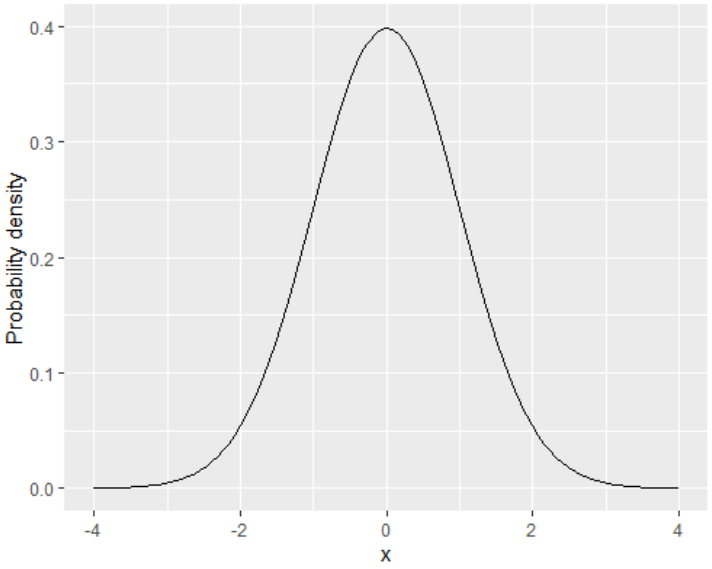
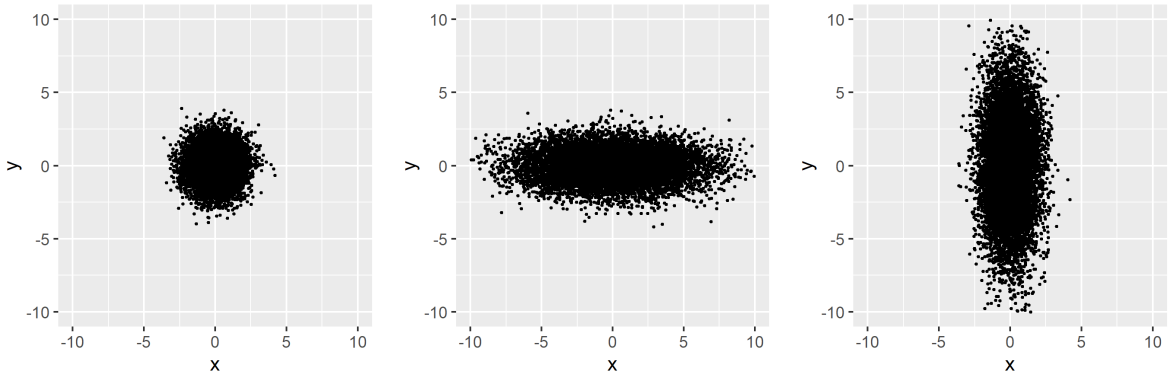


Figure 3.2: Normal distribution generated with mean 0 and standard deviation 1.

Where σ_X^2 is the variance in X, σ_Y^2 is the variance in Y, and the pair of off-diagonal elements $\sigma_X\sigma_Y$ and $\sigma_Y\sigma_X$ are the covariances between X and Y, and have the same value.

The shape of the cloud of points can be manipulated by changing the values in the covariance matrix (See Figure 3.3 and Figure 3.4 below for a graphical explanation). The properties of synthetic data that can be controlled in part by the covariance matrix are the distribution, elongation and orientation.

The placement of the cluster in two-dimensional space can be controlled by defining the mean vector at the initial cluster generation stage. Alternatively, the cluster can be moved after its generation by basic matrix operations (e.g. add 2 to each X variable).

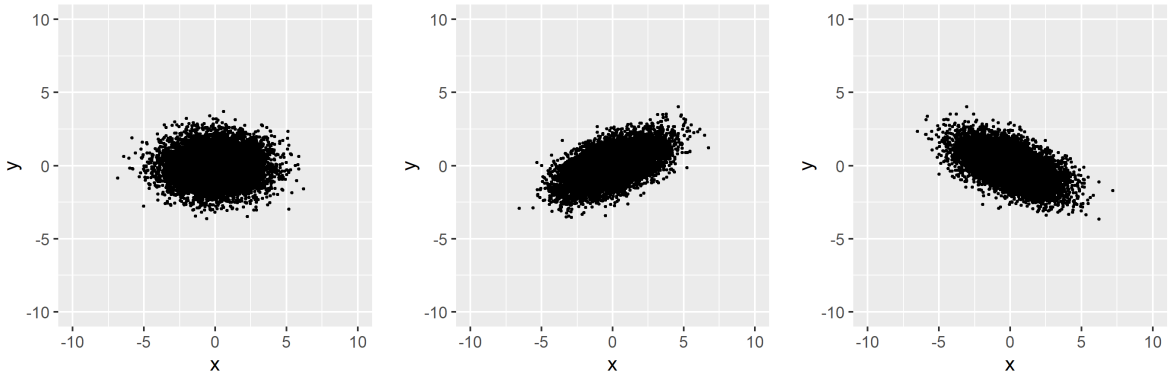


(A) $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ (B) $\Sigma = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}$ (C) $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}$

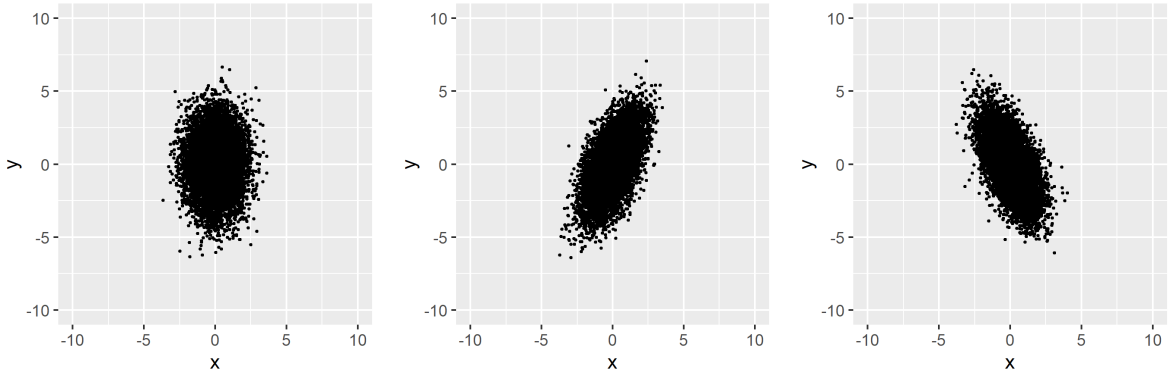
Figure 3.3: Data with a multivariate normal distribution can be controlled by changing the values in the covariance matrix. In this two-dimensional example, the means of both variables X and Y are 0. In plot (A), the variance of X (top right matrix element) is 1 and the variance of Y (bottom left matrix element) is 1, resulting in a circle shape. In plot (B), the variance of X has increased to 10, resulting in an ellipse elongated along X. In plot (C), the variance of X remains 1 and the variance of Y has now increased to 10, resulting in an ellipse elongated along Y. In all three plots, the covariances (off-diagonal matrix elements) are 0, indicating no relationship between X and Y.

3.2.4.2 Multi-cluster simulation

To create a synthetic dataset with two or more clusters, one method is to first generate each individual cluster, then combined all the rows of the separate cluster data tables together. A new variable to assign membership of each cell event to a cluster is helpful to sort, filter and make adjustments to clusters after they have been combined.



(A) $\Sigma = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}$ (B) $\Sigma = \begin{bmatrix} 3 & 1 \\ 1 & 1 \end{bmatrix}$ (C) $\Sigma = \begin{bmatrix} 3 & -1 \\ -1 & 1 \end{bmatrix}$



(D) $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}$ (E) $\Sigma = \begin{bmatrix} 1 & 1 \\ 1 & 3 \end{bmatrix}$ (F) $\Sigma = \begin{bmatrix} 1 & -1 \\ -1 & 3 \end{bmatrix}$

Figure 3.4: Covariance matrix values affect the distribution of the data cluster. Top row: in the horizontal ellipse in plot (A), the variance of X is 3, the variance of Y is 1, and there is no covariance relationship between X and Y. Plot (B) introduces a covariance of 1, resulting in a positive correlation between X and Y. The covariance is changed to -1 in plot (C), resulting in a negative correlation between X and Y. Bottom row: for comparison, the variances of X and Y are switched compared to the top row. The effects on the direction of the ellipse can be seen, as a result of the relative strengths of the correlation between X and Y.

With two or more clusters in a synthetic dataset, the location of each cluster can be controlled, and placed in overlapping or well-separated positions. This causes a need to measure the distance between clusters. Real flow cytometry data contains cell populations with varying degrees of separation. Manual gating can be performed with better accuracy and repeatability when the gap between cell populations is wide. But when cell populations are overlapping and merged, gating becomes difficult and the convention is to use negative unstained and fluorescence minus one (FMO) controls to determine where gates should be set [170]. Although there is currently no metric (nor a need for one) in flow cytometry to quantify the distance between gated cell populations, within this thesis, the quantification of the gap between clusters in synthetic data enhances their benchmarking utility, especially when analysing the performance of cell population separation features of automated data analysis software.

3.2.5 Description of the Separation Index

The separation index (SI) is used throughout this research to define the distance between clusters. The SI measures the magnitude of the gap between a pair of clusters based on the upper and lower percentiles of the two clusters [171]. In the one-dimensional example (Figure 3.5), the SI can be summarised as (3.3):

$$SI = \frac{L_2(\alpha/2) - U_1(\alpha/2)}{U_2(\alpha/2) - L_1(\alpha/2)} \tag{3.3}$$

where $L_i(\alpha/2)$ and $U_i(\alpha/2)$ are the sample lower and upper $(\alpha/2)$ quantiles of cluster i. The interpretation of the SI is relatively straight forward, the range is $[-0.999, +0.999]$ with values approaching $+1$ indicating increasing separation, SI of 0 indicating clusters touching, and SI approaching -1 indicating total overlap. In practice, the working range for the SI was $[-0.3, +0.3]$. These limits were defined because at a SI of $+0.3$ clusters were already very well separated, and at a SI of -0.3 clusters appeared well overlapped or merged.

3.2.6 Separation dataset generation

A library of two-cluster synthetic datasets in two dimensions with 1,000 datapoints per cluster was designed as an exemplar size of cell populations in real flow cytometry data.

The datasets were prepared using the R package *clusterGeneration* [172] because of the functions available for generating clusters with specified degree of separation. The following parameters were used:

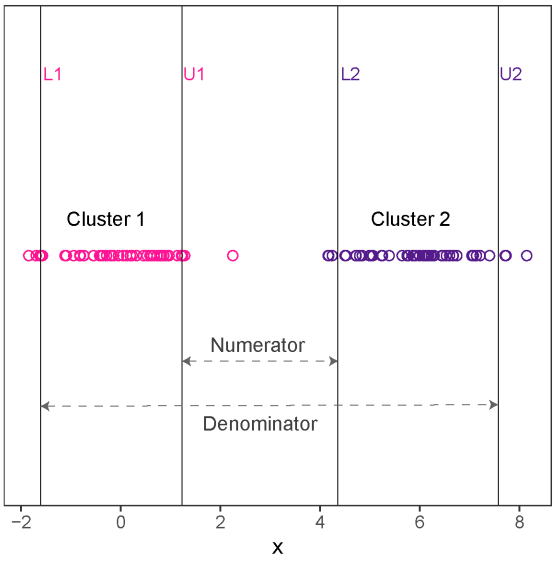


Figure 3.5: One-dimensional example of the separation index (SI) that measures the magnitude of the gap between two clusters. Vertical lines indicate the lower and upper quantiles of the clusters. The difference between U1 and L2 (numerator) is divided by the difference between L1 and U2 (denominator) to calculate the SI value. This method is robust against outliers in between the two clusters that may affect the SI. Figure adapted from [171].

- Number of clusters: 2
- Cluster size: 1,000 points per cluster
- Separation index (SI) values: from -0.3 to +0.3, at 0.1 intervals
- Cluster covariance matrices: eigenvalues between 1 to 5

This approach generated datasets with different degrees of separation between neighbouring clusters ranging from well-separated to merged. Nine random normally distributed cluster replicates were generated at each SI value. Covariance matrices were randomly generated from eigenvalues between 1 and 5 to give a variability in the diameter and shape of clusters that is similar to those seen in real flow cytometry data. These parameters produced clusters with known separation, but which were random in their elliptical shape attribute. Datasets were converted to FCS 3.1 format using the R package *flowCore* [134] to enable compatibility with flow cytometry specific software packages.

3.2.7 Skew dataset generation

A library of two-cluster synthetic datasets in two dimensions with 1,000 datapoints per cluster was designed, with different levels of skew and skew-direction pairs.

Skew datasets were built in multiple stages. First, individual skew clusters were prepared with the function `rmsn` in the R package *sn* [173], using the following parameters:

- Number of clusters: 1 (clusters later joined together)
- Cluster size: 1,000 points per cluster
- Mean vector: $[0, 0]$
- Covariance matrix: values between 1 to 5
- Skew parameter (α): values between 2.5 to 10, at intervals of 2.5

of which the skew parameter (α) regulated asymmetry. Likewise random cluster replicates were generated at each skew direction (left and right) along the x-axis.

During cluster generation, it was found that applying the skew parameter (α) caused the diameter of the elliptical cluster to reduce along the x-axis. To compensate for this, clusters were elongated to obtain a pre-skew diameter using the R package *rescale* [174]. The skewness of the clusters before and after rescaling were identical (measured using the R package *psych* [175]) determined by the asymmetry around the mean remaining unchanged (Figure 3.6A).

Next, two clusters were joined together, and one cluster shifted further away from the other through vector arithmetic operations in R (Figure 3.6B). The distance between two clusters was measured with the R package *clusterGeneration* [172], datasets with a SI value between -0.25 and -0.15 were selected for further processing because these were in the critical region around the SI value of -0.2 where software performances began to differentiate from each other.

A new level of complexity was introduced compared to normally distributed clusters because asymmetric clusters could be orientated in three ways: head-to-head, head-to-tail, and tail-to-tail (assuming the skew is introduced only along the x-axis). Clusters with the same α skew values were paired together (i.e. clusters with different skews were not combined).

Files were converted to FCS 3.1 standard using *flowCore* [134] and visualised within FlowJo software.

3.2.8 Real datasets

All real cell material used for comparison purposes was obtained with the approval of and in accordance with the respective Ethics Committees of Loughborough University and LGC, and under jurisdiction of the Human Tissue Authority. Two real cell datasets were used, each being representative real-world counterparts to the synthetic separation and

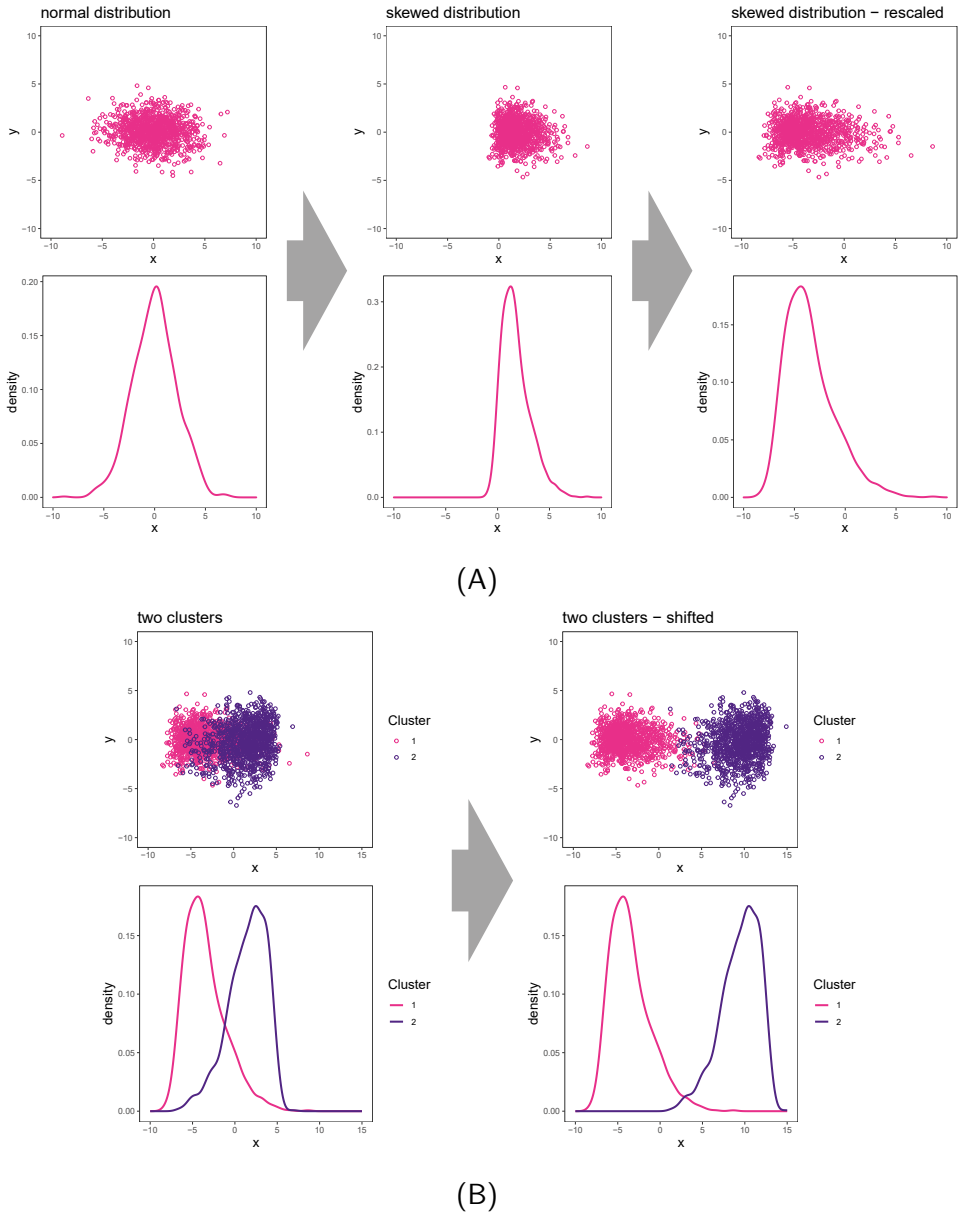


Figure 3.6: Workflow for skew dataset generation. Top panels show scatterplots, bottom panels show density estimates. (A) A cluster with a normal distribution is generated, then skew is added through the alpha parameter in the R package *sn*, then the cluster is rescaled. (B) Two clusters are combined, then the distances between them can be varied through vector arithmetic operations.

skew datasets, respectively: PBMC dataset 1 and 2. The datasets used were pre-existing and the Author did not generate the biological samples themselves.

3.2.8.1 Real cell PBMC dataset 1

Fresh whole blood from a healthy donor (Cambridge Bioscience, UK) was processed using Ficoll-Paque (Fisher) to isolate the buffy coat layer containing peripheral blood mononuclear cells (PBMCs). Cells were single-stained separately with CD4-PerCP-Vio700, CD45RO-APC-Vio770, and CCR7-VioBlue (all from Miltenyi Biotech). Data were acquired using a BD FACSCantoII cytometer equipped with 3 lasers (405nm/ 30mW, 488nm/ 20mW, 633nm/ 17mW). 100,000 cell events were collected. No compensation was performed for these single-stained samples. Pre-gating was applied to the lymphocytes population as depicted in Figure 3.7 before the use in comparative analysis.

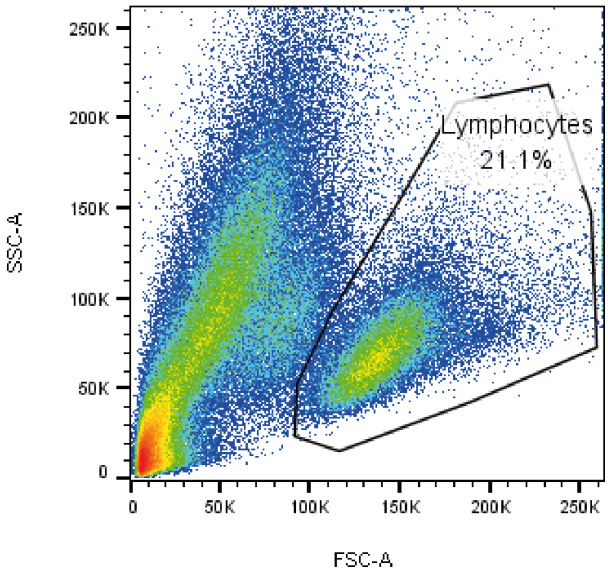


Figure 3.7: Example of pre-gating on lymphocytes applied to PBMC dataset 1.

3.2.8.2 Real cell PBMC dataset 2

PBMCs (LGC, UK) were stained with CD3-BB515, CD4-BB700, CD8-APCH7, CD45RA-BV786 (all from BD Biosciences), and live/dead fixable aqua dead cell stain (Invitrogen). Data were acquired using a BD LSRFortessa cell analyser equipped with four lasers (355nm /20mW, 405nm/ 50mW, 488nm/ 50mW, 640nm/ 40mW). 200,000 cell events were collected. Single-stained beads and fluorescence-minus-one controls were used to calculate compensation. Pre-gating was applied to the dataset as depicted in Figure 3.8 to isolate skewed cell populations.

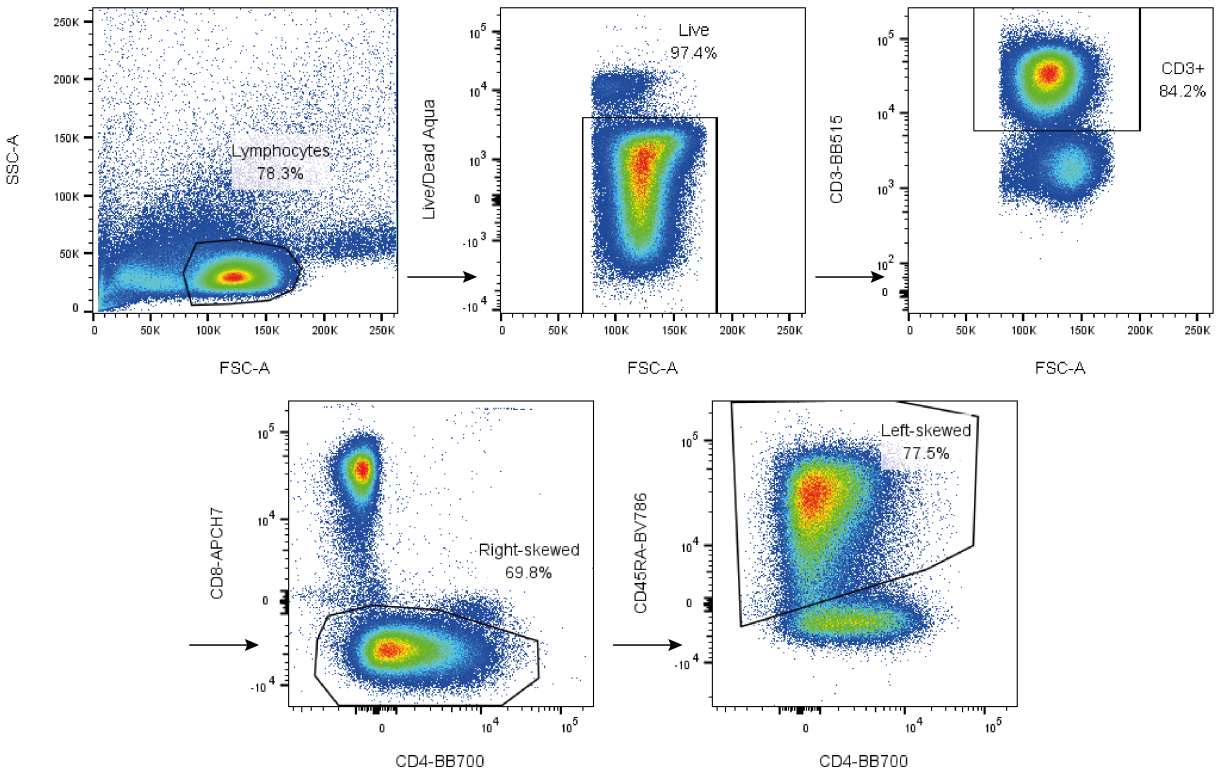


Figure 3.8: Example of pre-gating applied to PBMC dataset 2 to isolate skewed populations.

3.2.9 SPADE3 analysis of synthetic datasets

The MATLAB-based SPADE3 implementation with graphical user interface was run within MATLAB R2019a. Each FCS file was run separately. User input parameters that were selected were: overlapping markers used for SPADE tree = CH1, CH2; ignore compensation; no transformation. All other settings were left as default values (local density neighbourhood size = 5, local density approximation factor = 1.5, maximum allowable cells = 50,000, outlier density = 1, target density = 20,000 cells, algorithm = k-means, number of desired clusters = 100).

3.2.10 Statistics and performance metrics

The software outputs were recorded, and the absolute difference between the cell count of cluster 1 to the reference value was calculated as:

$$\text{Absolute difference to reference count} = |A - B| \tag{3.4}$$

where A is the reference cluster 1 count, and B is the software cluster 1 count.

Methods used for statistical analysis included the mean, sample standard deviation (SD), coefficient of variation (CV), and metrics derived from the confusion matrix as

shown in Table 3.4.

Table 3.4: Confusion matrix

	Reference	
Predicted	Target	Non-target
Target	True positive	False positive
Non-target	False negative	True negative

As in binary classification [176], here a true positive (a 'hit') is defined as the correct SPADE3 assignment of a target cell to its reference target population set during cluster generation. Events in cluster 1 of the synthetic datasets were arbitrarily selected as the 'target' cases. SPADE3 assignment of a non-target cell to its non-target population is a true negative. Misclassification of a non-target cell to a target population is a false positive, and misclassification of a target cell to a non-target population is a false negative (a 'miss'). The evaluation metrics calculated from the confusion matrix include the accuracy (Eq. 3.5), precision (Eq. 3.6), recall (Eq. 3.7) and F1 measure (Eq. 3.8) [177]. The individual cell assignments to a cluster predicted from SPADE3 was compared with the reference cell assignments, using the R package *caret*.

$$\text{Accuracy} = \frac{\text{True positive} + \text{True negative}}{\text{Total population}} \tag{3.5}$$

$$\text{Precision} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}} \tag{3.6}$$

$$\text{Recall} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}} \tag{3.7}$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.8}$$

3.3 Results

In flow cytometry, data typically contain cell populations that are positive or negative for a marker of interest. The distance between the positive and negative cell populations is variable, ranging from well resolved to merged. Multiple factors affect this separation, including but not limited to biological attributes (the level of marker expression, affinity

and avidity of antibody binding, the number of antibody bound per cell) and assay variables (antibody concentration used in the staining process, fluorophore brightness and dye stability). This separation directly impacts the accuracy and repeatability of manual data analysis, with significantly higher technical variation seen in poorly resolved populations compared with clearly defined cell populations in human peripheral blood [178].

3.3.1 Distance between clusters

To simulate flow cytometry data with different distances between a positive and a negative population, a normally distributed synthetic two-cluster dataset with different degrees of separation between clusters was generated. For comparison, the SI between two clusters in a real cell dataset was measured. The PBMCs dataset 1 contained negatively and positively stained populations in each fluorescent channel. These subpopulations were separated using the automated cell population identification software SPADE3 [105]. Then the magnitude of the gap between pairs of real cell clusters along each channel was measured using the `sepIndex` function in the *clusterGeneration* package. Similar SI values were observed between the real-world positive and negatively stained cell populations and the synthetic clusters, within the range of -0.3 and $+0.2$ (Figure 3.9). These results show the method defined here for generating synthetic flow cytometry datasets is able to successfully simulate the distance parameters seen between clusters in a real example of flow cytometry data.

It is noted that the total size of the synthetic datasets (2×10^3 events) are smaller than the ones in the PBMC dataset 1 used (10^5 events), because it was not the intention here to represent similar cluster size properties, but rather the gap that exists between the clusters, which can be the same distance for both large or small clusters. The reader is referred to Chapter 5, where varying cluster sizes are investigated, for synthetic datasets with up to 10^6 total events.

3.3.2 Clusters with non-normal distributions

The synthetic datasets generated in section 3.3.1 contain clusters following a normal distribution, visualised as symmetrical bell curves for univariate data or symmetrical circles and ellipses in scatterplots for multivariate data. However, real flow cytometry data consist of cell population clusters that follow a normal distribution as well as those that display non-normal distributions. The exact distribution along a marker channel is difficult to predict and may depend on the state of the cell along a differentiation pathway. For example, a stable haematopoietic stem cell population may display a normal

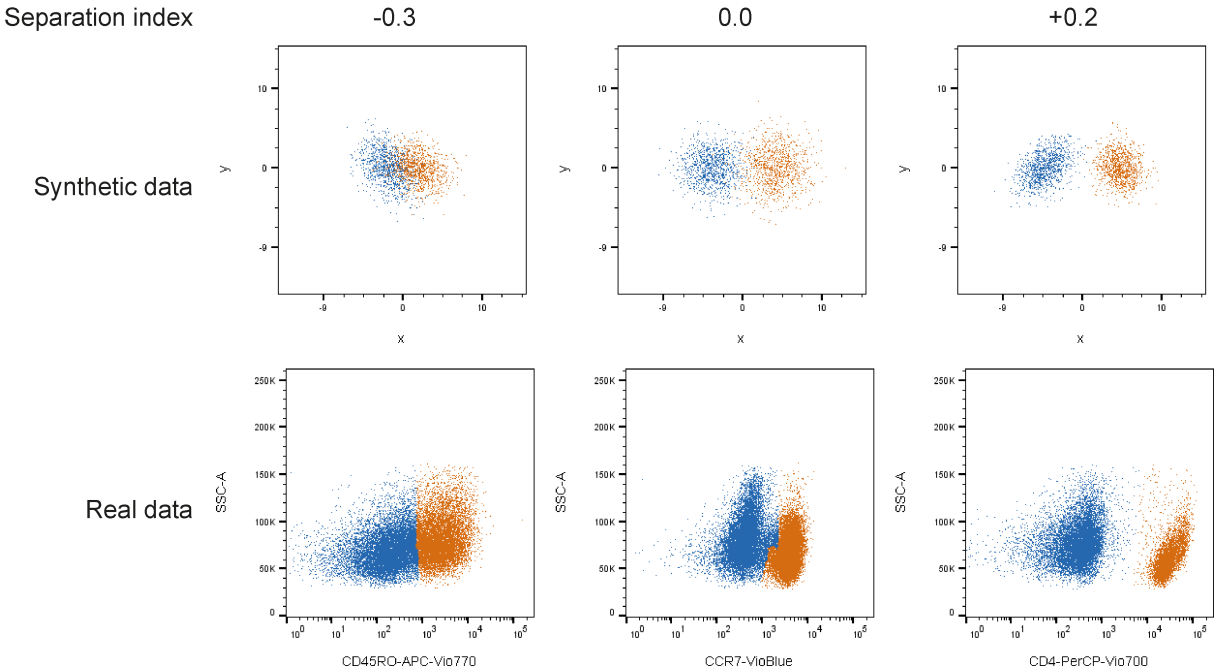


Figure 3.9: Comparison of synthetic and real clusters with representative separation index values ranging from -0.3, 0.0 to $+0.2$, showing overlapping, touching and well-separated clusters, respectively. Top panel shows synthetic data generated from R package *clusterGeneration*, bottom panel shows real cells (PBMC dataset 1) after automated population detection and partition with SPADE3 software followed by separation index calculation.

distribution of CD34+ expression that transitions to a non-normal distribution during cell differentiation as CD34 expression decreases [179].

Non-normal data are characterised by asymmetry around the sample mean. These cell populations can display positive (right) skew or negative (left) skew. The skewness can be estimated using the adjusted Fisher-Pearson coefficient of skewness (p value) [180], where a normal distribution has a skewness value of $p = 0$, a positive skewness value indicates a tail pointing to the right, and a negative skewness value indicates a tail pointing to the left. The further away the value is from 0, the greater the skew and typically the longer the tail.

There are different strategies to generate synthetic flow cytometry datasets with skewed cell populations. One method is to create multiple Gaussian distributions that can then be merged together to form an overall distribution with the desired skew. This strategy has been used previously to create synthetic data with non-convex shapes [71, 83]. This method may require many rounds of trial and error. To avoid this shortcoming, here a different method using the *sn* R package was developed and tested to generate random clusters with multivariate skew-normal distributions [181].

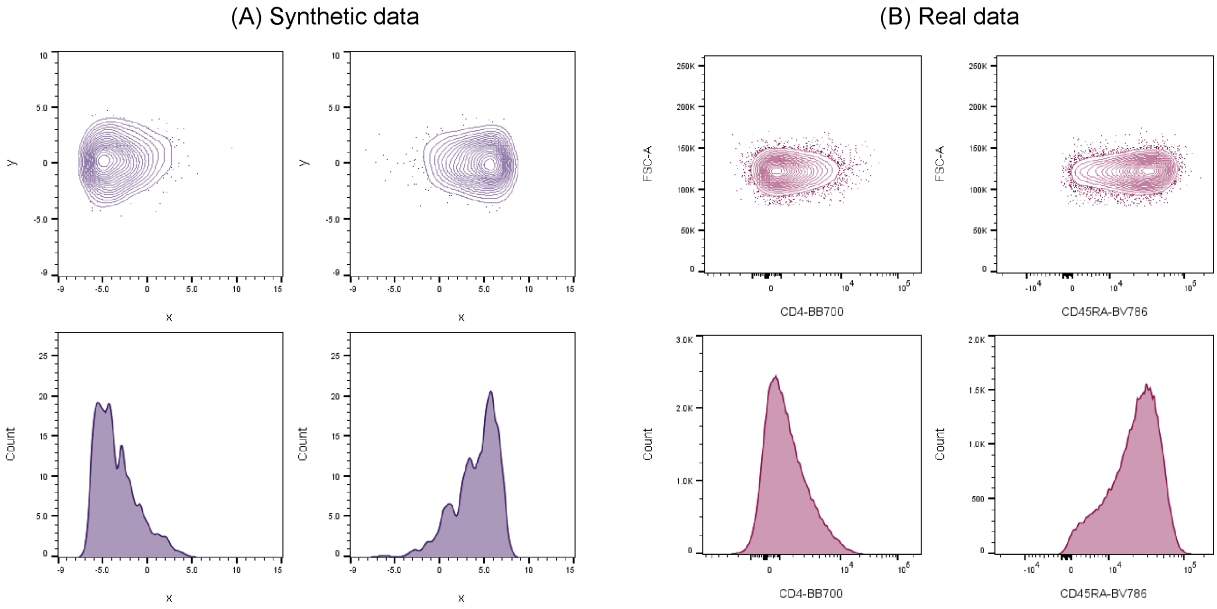


Figure 3.10: Comparison of (A) synthetic and (B) real cell (PBMC dataset 2) flow cytometry data with skewed distributions. Both left-skewed and right-skewed synthetic clusters can be generated that mimic real data. Contour plots used rather than dot plots for better visualisation of skewed distributions. Asymmetry around the mean is clearly shown with contour plots (top) and histograms (bottom).

The multivariate skew normal distribution extends the class of normal distribution (defined through a mean vector and covariance matrix) by the addition of a skew parameter. A visual comparison of the skewed clusters generated from computer simulation against real cell populations from the PBMCs dataset 2 demonstrates that the synthetic and real cases are comparable (Figure 3.10). This result shows that the simulated data generated here is a realistic model of both positive and negative skew observed in real flow cytometry cell populations, and therefore has biological relevance. Thus, the synthetic dataset can be reliably used to gain understanding on how automated software responds to skewed flow cytometry data, with the additional benefit of the ability to systematically control the strength of the skew as well as the absolute cell number.

The strategy devised here to create a dataset with multiple skewed clusters was to generate individual clusters in parts then combine them together to form one whole dataset. The gap between the clusters can be controlled by shifting one cluster closer or further away to the other through vector arithmetic operations, with the SI being measured after the clusters were combined. With skewed clusters, a new level of complexity is introduced compared to normally distributed clusters, because assuming the skew is introduced only in one parameter, each cluster can be left-skewed or right-skewed. Thus, the possible

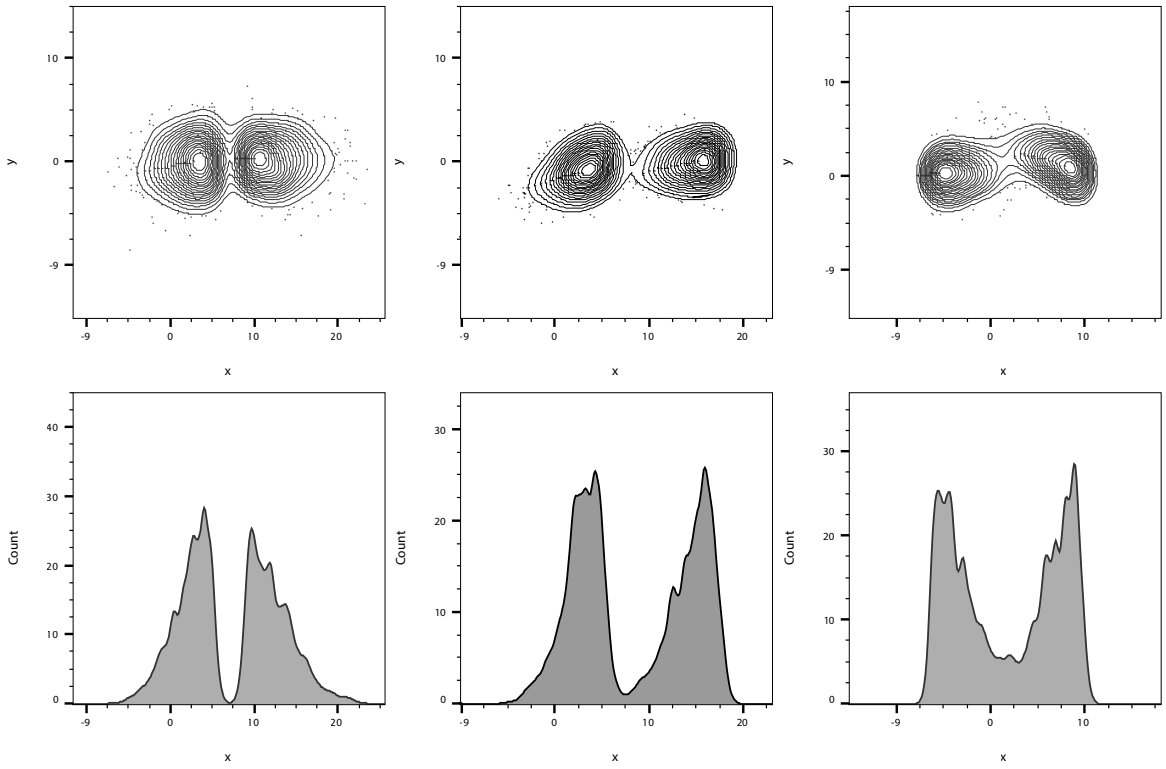


Figure 3.11: Generation of synthetic two-cluster skewed datasets. Three combinations of cluster pairs are shown here; head-to-head, head-to-tail and tail-to-tail.

permutations of pairs of skewed clusters in a two-cluster dataset increases from one to three. In this research, these combinations are referred to as: head-to-head, head-to-tail, and tail-to-tail (Figure 3.11).

3.3.3 Application of synthetic datasets to a cell population identification software

To demonstrate the efficacy of the synthetic datasets, they were passed through an exemplar software, SPADE, in order to illustrate how synthetic data can reveal limitations of automated software, and can provide a deeper understanding of the inner workings of 'black box' algorithms in a way that real cell datasets are unable to. SPADE was selected because the work in Chapter 2 identified it as a highly cited software tool. SPADE (spanning-tree progression analysis for density-normalised events) is a software package that uses automated down-sampling, clustering and minimum spanning tree construction to aid analysis of high-dimensional flow cytometry data [68]. There are two versions of SPADE with different algorithms. The original SPADE1 applies a stochastic down-

sampling algorithm paired with an agglomerative hierarchical clustering algorithm that produces different outputs when run on the same data [105]. This specific issue of reproducibility in SPADE1 was subsequently resolved in SPADE3 by removing the stochastic algorithms and replacing them with deterministic ones. In addition, a tree partitioning function was introduced to assist interpretation of the outputs. On the basis of these improvements to the reproducibility and functionality of the software, the SPADE3 version was used in this Chapter rather than SPADE1.

SPADE3 was used to process synthetic datasets using default parameters (as described in Section 3.2.7). The auto tree partitioning tool was used to split the spanning tree into two populations, then the population number was compared to the known reference value of 1,000 cells per cluster, or 50% of total cells events, for both the separation and skew datasets. For the separation dataset, the absolute difference in cell count of each cluster between the software output and the reference value was calculated for each SI condition. The results show that the accuracy and repeatability of SPADE3 decreased as the SI decreased from $+0.3$ to -0.3, with performance deteriorating noticeably at a SI value of -0.2 and below (Figure 3.12). These results were to be expected, because defining the boundary between one cluster and another becomes progressively more difficult as clusters get closer together.

The benefit of applying the synthetic datasets to test software such as SPADE3 was the ability to quantify for the first time the SI value where the software began to lose performance. The high level of control in designing the gap between cell populations within the synthetic datasets would have been very difficult to achieve with real cell data. Furthermore, since the absolute counts and frequencies of each cell population was known in the synthetic dataset, the evaluation of the software was based on robust absolute traceable figures, and did not rely on comparison with a manually gated reference subpopulation count, which has already been shown to be operator dependent and potentially biased [42, 43].

In the design of the skew dataset, a constant SI value of -0.2 between clusters was chosen because it fell in the critical region where the SPADE3 software began to deteriorate. The skew dataset was processed through SPADE3, then the difference in cell population percentage of the cluster between the SPADE3 output and the reference value was calculated for each skew condition and cluster pair orientation. It was found that, for each cluster pair orientation, increasing the level of skewness in the clusters had no effect on the accuracy and repeatability of SPADE3. However, at each level of skewness, SPADE3 was able to partition the two clusters with improved performance when the orientation was tail-to-tail, followed by head-to-tail and finally head-to-head (Figure 3.13).

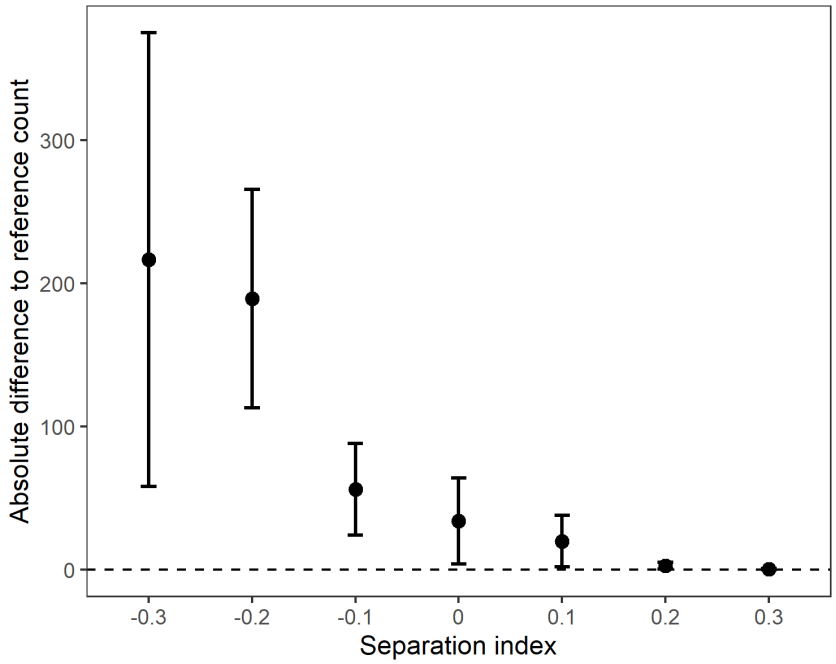


Figure 3.12: The synthetic two-cluster separation dataset was run through SPADE3, then the absolute difference between the SPADE3 cell count to the reference value of cluster 1 was calculated. Result demonstrates the accuracy and repeatability of SPADE3 deteriorates as the distance between clusters decreases. Data shows mean ±1SD.

This pattern of performance appeared to correlate with the density of points between the two clusters. The skew dataset was planned with skewness and skew orientation among the variable design factors, and the separation between clusters as constant factors. The systematic way this skew dataset was designed allowed for the pattern of behaviour of SPADE3 to become apparent. This finding suggests the SPADE3 algorithm is well suited to analysis of skewed data albeit with a performance bias and sensitivity depending on skewed cluster orientation. This may not be the case for other algorithms that use different clustering techniques, in particular those that use a Gaussian mixture model-based clustering approach. Further work to investigate this in a software comparison study is conducted in Chapter 4.

3.3.4 Assessment of software performance based on synthetic data

One of the benefits of synthetic data is that, as well as 'true' population counts and frequencies, an estimate of the true membership of a cell to its cluster is known *a priori*. This is not the case with real cell data, where membership of a cell to a population is

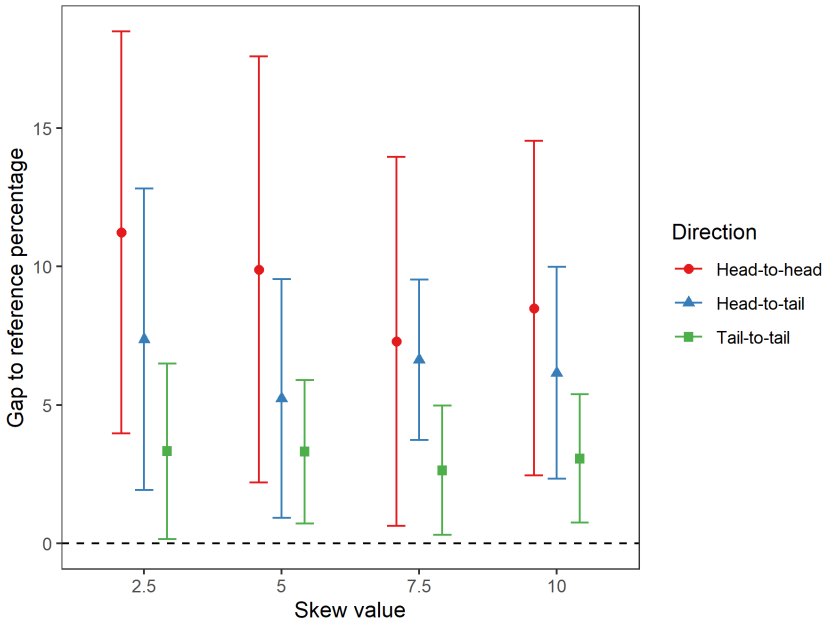


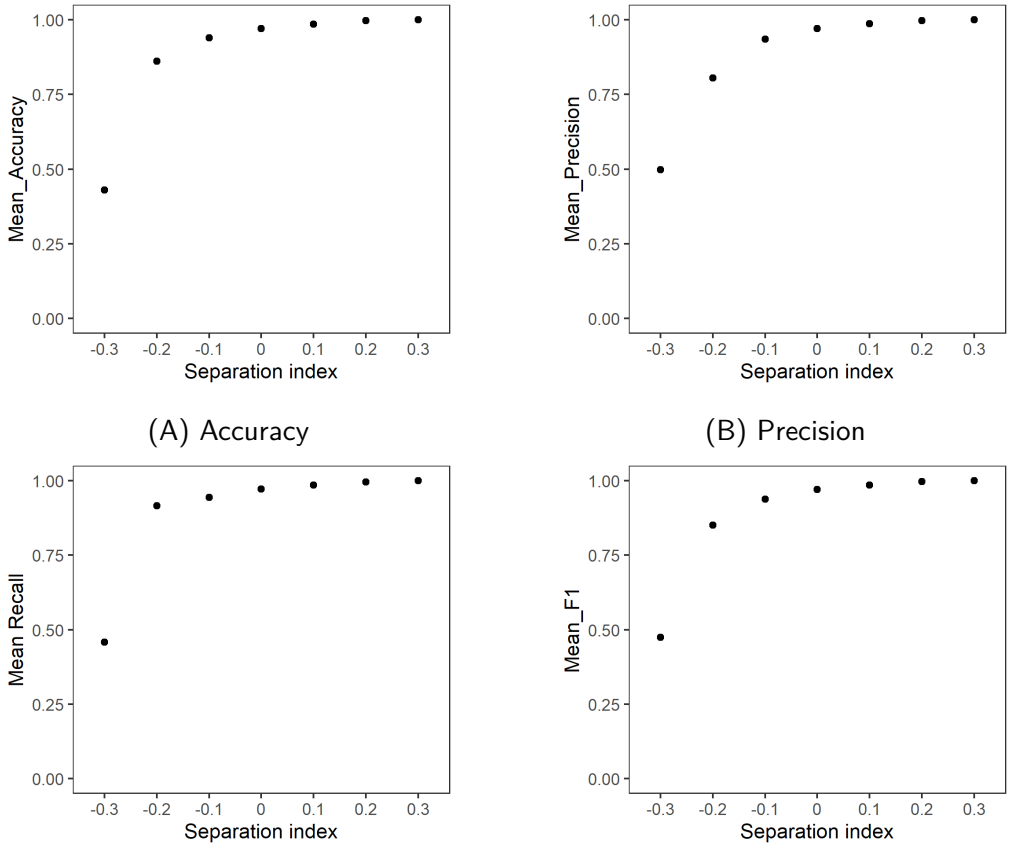
Figure 3.13: The synthetic two-cluster skewed dataset was run through SPADE3, then the gap between the SPADE3 cell population percentages and the reference cell population percentages was calculated. Increasing the cluster skewness in the datasets did not affect SPADE3 performance. However, the accuracy and repeatability of SPADE3 improved as direction of the cluster pairs changed from head-to-head to head-to-tail, and then to tail-to-tail. All clusters had separation index of -0.2. Data show mean $\pm$1SD.

estimated by an analyst performing manual gating. Here, the evaluation of robust performance metrics of SPADE3 runs on synthetic datasets was demonstrated using confusion matrix analysis.

Each cell event in the synthetic dataset was pre-assigned a cluster membership on generation. These cluster memberships were withheld for the SPADE3 analyses. After running the datasets through SPADE3, the software predictions of cluster memberships for all 2,000 cell events were compared with the reference cluster memberships using the R package *caret*. Events in cluster 1 were arbitrarily assigned as positive cases.

The results from the SPADE3 analysis of the synthetic separation dataset (Table 3.5) showed a classification accuracy (Eq. 3.5) greater than 90% with SI values of -0.1 or greater. Accuracy fell to 86% and 43% at SI values of -0.2 and -0.3 respectively (Figure 3.14A).

The same pattern appeared with the precision metric, also called positive predictive value (Eq.3.6) with values greater than 90% at SI values of -0.1 or larger, then falling to 81% and 50% at SI values of -0.2 and -0.3 respectively (Figure 3.14B). The recall metric



(A) Accuracy (B) Precision

(C) Recall (D) F1

Figure 3.14: Performance metrics for SPADE3 analysis of separation dataset.

Table 3.5: Performance metrics for SPADE3 analysis of separation dataset

Separation index	**Mean Accuracy (%)**	**Mean Precision (%)**	**Mean Recall (%)**	**Mean F1 (%)**
−0.3	43.0	49.8	45.8	47.4
−0.2	86.1	80.5	91.6	85.1
−0.1	93.9	93.5	94.4	93.9
0	97.0	97.0	97.2	97.0
0.1	98.6	98.7	98.5	98.6
0.2	99.7	99.8	99.7	99.7
0.3	99.98	99.97	100.00	99.98

that measures the rate of true positives identified (Eq. 3.7) gave perfect scores of 100% at SI of +0.3, scored greater than 90% at SI values of −0.2 or greater, and deteriorated

to 46% with SI of −0.3 (Figure 3.14C). The F1 score (Eq. 3.8) was calculated from the harmonic mean of precision and recall to give the overall accuracy of SPADE3. The F1 score remained above 90% for SI values greater than −0.1, then fell to 85% at a SI value of −0.2 and reduced further to 47% at a SI value of −0.3 (Figure 3.14D).

The results from the classification analysis reinforce the finding that SPADE3 performs strongly when clusters are well-separated, but the accuracy decreases as clusters approach a SI value of −0.2 and falls to below 50% when processing overlapping clusters with SI values of −0.3. The application of this synthetic flow cytometry dataset in this instance has helped to illustrate good performance characteristics and ranges, but also limitations of SPADE3 with respect to cluster separation both with normal and skewed cluster probability distributions.

When taken together in comparison to reporting the accuracy and precision expressed as the difference to reference count ±1SD (as in Figure 3.12), the binary classification metrics calculated here were judged to be more difficult for users to interpret, particularly without any given acceptability thresholds. For example, for the separation dataset analysis at an SI of −0.1, a mean F1 score of 94% is less informative than a difference to reference count of 56 ± 34, and such a high score could potentially mislead users into assumptions of high performance and a false sense of confidence in the measurement outputs of software tools. For this reason, the classification metrics have not been used in Chapter 4, where similar separation and skewed synthetic datasets are applied. However, these metrics were explored in Chapter 5 for the rare cell datasets.

3.4 Discussion

This Chapter has introduced a systematic method of designing and generating synthetic flow cytometry datasets, with specific focus on control of the distance between clusters and the probability distributions of events within clusters. The computer-generated flow cytometry datasets have been applied to an automated data analysis software, SPADE3, and results have shown that the synthetic datasets are capable of critically assessing the quality of the software outputs and hence the software performance. In addition, this Chapter has given an example of quantifying performance assessment using synthetic datasets that is robust compared with using real-world datasets.

The systematic approach implemented within this research to produce flow cytometry datasets is straight forward to execute computationally, but would be complicated to achieve experimentally due to uncontrollable external sources of variation within real cell datasets, thus synthetic datasets here overcome the limitations of acquiring real datasets.

The synthetic datasets have the same range of data properties as their biological equivalents and can serve as credible substitutes for real flow cytometry datasets for the testing of automated cell population identification software.

It is noted here that in most cases algorithms that underpin flow cytometry analysis software have been previously published, but it is (at times) inherently opaque implementation of algorithms in executable code that whilst allows understanding of inputs and outputs, does not allow a full understanding of the data transfer functions. Application of these synthetic datasets to an automated cell population identification software such as SPADE3 can therefore help users understand how the underlying 'black box' algorithm works.

The work presented here identified the regions where SPADE3 began to lose performance, specifically where two clusters are located at a SI value of -0.2 or less. These results suggest that SPADE3 is not specifically affected by the probability distributions of data, but is more sensitive to the relative density of data points between two clusters.

Findings such as these can provide guidance to users on software selection when having to contend with large array of potential software solutions [167] (i.e. in this exemplar whether SPADE3 would be an appropriate tool of choice for automated analysis of real data containing heavy overlapping of clusters), and then help to understand the validity of their automated analysis outputs.

A further benefit of synthetic datasets was apparent when assessing the accuracy and repeatability of software performance, because the synthetic datasets contained an estimate of the true assignment of cells to clusters designed into the data, therefore allowing the comparison of software predictions with known 'true' conditions hence an absolute analysis. In contrast, with real data, the assignment of cells to subpopulations must first be manually determined (often with potential difficulty and error), then the dataset labelled, before comparisons to software outputs can be calculated. These additional steps are time-consuming and less robust.

The variability observed in manually gated datasets means either the analysis from a single expert must be taken as the best estimate of the 'true value', or a pooled manual analysis from a group of experts is used. The first option risks bias, and the second is dependent on the accuracy and repeatability of the group. In both instances, it is difficult for the final analysis to be as robust compared with synthetic datasets.

There are also a few disadvantages with synthetic datasets. Since synthetic datasets are built on examples of real data, the quality of real data will directly impact that of synthetic ones. For this reason it is valuable to have access to reliable sources of flow cytometry experimental data that are acquired following MIFlowCyt standards [50].

Constructing mathematical models of the phenotypic properties of cell populations in flow cytometry data relies on having large quantities of those cell subsets of interest, that are well characterised. This task may be straightforward for stable cell lines or normal healthy donors, however it is more difficult to achieve in disease cases, where occurrences in the general population are fewer in number. Therefore, the cell characteristics inferred from these limited real world examples may be less accurate. Additionally, the cell populations among disease cases may present wide biological variations in cellular phenotypes from patient to patient (e.g. heterogeneity in circulating tumour cells) and may be dependent on stages of disease progression. As such, although the aim is to create synthetic datasets that are as realistic as possible, there may be features missing as a function of boundary conditions and design assumptions.

An further difficulty with synthetic datasets is the need to validate their accurate representations of the properties of real data. In this research, this was done in a quantitative way for validating separation between clusters, and in a qualitative way for skew properties through visual comparisons, and the other data properties were not validated because it was outside the scope of this initial study. These different methodologies perhaps allude to the nascent nature of the field and the lack of established methods and tools available for extensive validation of synthetic datasets against real ones.

Besides being a benchmarking tool for software developers, possible further applications of synthetic datasets include their use as educational and training tools for manual gating, as part of external quality assessment (EQA) and proficiency testing schemes. In addition, as cell identification and quantification in medical diagnostics and cell therapy/ regenerative medicines manufacturing fields move increasingly towards automated machine learning and artificial intelligence techniques, it is likely that synthetic datasets will have important regulatory applications as digital reference materials and standards, as well as potential regulatory implications.

Following this Chapter's assessment of SPADE3 performance when challenged with separation and skewed synthetic datasets, the next chapter (Chapter 4) extends the research with a comparison study, using the same synthetic datasets, across multiple automated data analysis software tools that employ various clustering algorithms such as k-means, hierarchical, partition, density-based, model-based, spectral clustering and self-organising maps.

Further investigations on flow cytometry synthetic datasets will aim to generate datasets with controls on other flow cytometry data properties identified in Table 3.2, and more complex datasets with multiple controlled factors. Chapter 5 will focus on developing and optimising synthetic datasets with rare cell populations with and without skewed

distributions, and Chapter 6 will focus on noise characteristics.

3.5 Chapter conclusions

- The common characteristics and statistical properties of flow cytometry data were identified as the: number of clusters, number of datapoints, number of dimensions, cluster separation, cluster distribution, and noise.
- Using the R programming environment, a method was developed to simulate cluster separation and skewed distributions in synthetic datasets in a highly controlled manner.
- The simulated clusters with different degrees of separation were validated with real cell data showing a range of equivalent distances between clusters.
- The simulated skewed clusters were shown to mimic the non-normal distributions apparent in real PBMCs data with clear asymmetry around the mean.
- The synthetic datasets were successfully processed through SPADE3, with their known ground truths allowing SPADE3's performance to be defined with absolute statements of accuracy and repeatability.
- Synthetic datasets can overcome certain limitations of real datasets, such as difficulties in their acquisition in terms of cost and time, lack of ground truth, requirement for laborious labelling, privacy concerns and ethical considerations.

Chapter 4

Software comparison

The publication listed below was an outcome of the work reported in this Chapter:

> **Cheung M**, Campbell JJ, Thomas RJ, Braybrook J, Petzing J. Assessment of Automated Flow Cytometry Data Analysis Tools within Cell and Gene Therapy Manufacturing. International Journal of Molecular Sciences. 2022; 23(6):3224. https://doi.org/10.3390/ijms23063224

4.1 Introduction

In the previous Chapter, a method to generate synthetic flow cytometry datasets containing controlled separation between clusters with normal or non-normal probability distributions was developed. The synthetic datasets demonstrated clear similarities in cell distribution characteristics when compared against real world flow cytometry data (Figures 3.7 and 3.8), and were applied to an exemplar automated cell population identification tool, SPADE3, to assess its performance.

This Chapter brings together the work carried out in Chapters 2 and 3 by using these synthetic datasets to perform a study to assess the accuracy and reproducibility of the software tools selected at the end of Chapter 2, each of which implement different unsupervised clustering algorithms (Table 2.3).

4.1.1 Chapter aims

The aims of this Chapter are to:

- Apply systematically designed synthetic flow cytometry datasets for software comparison studies.

- Investigate the effect of cluster separation on software performance in terms of accuracy and repeatability.
- Investigate the effect of normal/skewed cluster distributions on software performance in terms of accuracy and repeatability.
- Understand the performance characteristics of software tools at their base functionalities, and uncover their potential strengths and limitations.
- Demonstrate the capability of synthetic datasets as reference datasets for benchmarking of software tools, towards achieving confidence in automated cell measurements.

4.2 Methods

4.2.1 Datasets

In order to perform a fair comparison between different automated data analysis software, synthetic flow cytometry reference datasets were designed and generated (as described in Chapter 3). Out of the commonly recognised data characteristics or potential statistical attributes identified (in Chapter 3, Table 3.2), the separation and the skew characteristics were targeted for controlled modification in the datasets, because these properties had not been addressed in previous work and/or the designs had not been approached in a systematic manner. To retain the focus on these properties, non-target characteristics such as cluster sizes and the number of dimensions were kept constant, and the element of noise relating to real data was excluded to keep the datasets clean.

4.2.1.1 Separation dataset

The purpose of these datasets was to evaluate software performance in identifying and partitioning cell populations as the clusters came close together. A two-cluster separation dataset was generated as described in Section 3.2.5. A three-cluster separation dataset was generated using an identical method, but with the number of clusters set to 3.

4.2.1.2 Skew dataset

The purpose of this skew dataset was to evaluate software performance in identifying and partitioning cell populations as the clusters displayed different levels of non-normal distributions.

Skew datasets were built as described in Section 3.2.6, generating a library of two-cluster synthetic datasets in two dimensions with 1,000 datapoints per cluster, with different levels of skew and skew-direction pairs (head-to-head, head-to-tail, and tail-to-tail).

4.2.2 Software runs

The synthetic datasets were processed through six flow cytometry automated data analysis software (Table 4.1), each of which implement different unsupervised clustering algorithms: Flock2 [78], flowMeans [75], FlowSOM [73], PhenoGraph [70], SPADE3 [68, 105] and SWIFT [88, 89]. The datasets were also processed through SPADE1 however the runs were unable to be completed successfully (error messages encountered mid-run).

The input parameters used for software runs are listed in Table 4.2. The same input parameters were used for both the separation and skew datasets. To enable comparability between software, user parameters were kept as similar as possible, and in most cases default settings were used. Where required, manual intervention of outputs was kept to a minimum.

4.2.2.1 Flock2

Flock2 analysis was performed on the web-based platform ImmPort Galaxy version 1.2 [182]. FCS files were uploaded to the platform, and converted to a text file using the 'Convert FCS to Text' tool prior to analysis.

4.2.2.2 flowMeans

Initial flowMeans analysis was performed on FlowJo v10 using the R-based plugin included with the installation package. However, later in this research, issues with the plugin began to emerge and after the FlowJo platform failed to deliver results, analysis was switched to the native R platform configuration of flowMeans. Visual inspection of the outputs from both platforms showed comparability with highly similar clustering characteristics.

4.2.2.3 FlowSOM

FlowSOM analysis was performed on the web-based platform ImmPort Galaxy version 1.2 [182]. FlowSOM (Galaxy Version 1.0) was run using a grid size of 3×3 from the default of 10×10, to force three clusters to be returned. A grid size of 2×2 was found to be an invalid input; two clusters could not be returned.

Table 4.1: Description of computational tools used in this study.

Computational tool	Description	Reference
Flock2	FLOw Clustering without K; grid-based density clustering algorithm, where the data are divided into hyper-regions, then dense regions are identified, merged and points assigned to their nearest centroids.	[78]
flowMeans	k-Means based clustering that allows multiple clusters to model a single population, with overlapping clusters later being merged.	[75]
FlowSOM	A workflow that reads the data, builds a self-organising map (SOM), builds a minimal spanning tree then computes a meta-clustering output.	[73]
PhenoGraph	Constructs a k nearest neighbour graph from high-dimensional data, then uses the Louvain community detection algorithm to partition the graph into sub-populations.	[70]
SPADE1, SPADE3	Spanning-tree progression analysis for density-normalized events; performs deterministic density-dependent downsampling, then k-means based clustering, followed by minimal spanning tree construction. A tree partitioning algorithm aids semiautomated interpretation of data.	[68, 105]
SWIFT	Scalable Weighted Iterative Flow-clustering Technique; Gaussian mixture model-based clustering, followed by splitting and merging steps to obtain final clusters that are unimodal but not necessarily Gaussian.	[88, 89]

4.2.2.4 PhenoGraph

PhenoGraph analysis was performed on the R platform using *Rphenograph* version 0.99.1 [183]. The output number of clusters is not a specifiable parameter in PhenoGraph, instead, the input parameter k (number of nearest neighbours) needs to be determined to cause the software to output an estimated desired number of clusters. Initial testing on synthetic data with 2,000 points found the default input of $k = 30$ returned outputs of approximately 16 clusters (Figure 4.1A). This output required excessive subjective manual intervention and interpretation to reduce down to two or three clusters, so could not be used. To reduce the output number of clusters, the input k value needed to be increased. However, increasing k until the output reached two or three clusters was not practical because of long run times (>5h) required to compute k nearest neighbours of each data point. Subsequent testing showed a starting k value of 150 returned manageable outputs of approximately eight clusters (Figure 4.1B). If the output remained above eight clusters, k was increased by 50 iteratively until the output reached eight clusters or fewer.

It is noted that the PhenoGraph algorithm is intended for clustering of high-dimensional data and not necessarily optimised for analysis of the two-dimensional datasets applied here. From our groups broader collaborations recently and over many years with; big pharma, contract manufacturing organisations, clinical centres, external quality assessment (EQA) centres, international measurement institutes etc, it is very apparent that the availability of (in this instance) flow cytometry data analysis software can often lead to inappropriate application of said software. This is not unique to biometrology or the biosciences. This behaviour is repeated in other metrology domains and in other industrial sectors. It is a human factors issue and is symptomatic of operators trying to glean further insight from data when not necessarily understanding the boundary conditions and performance criteria of the software tools that are easily and commercially available. It is often the case that available functions within the software solutions can be too comprehensive for tasks at hand. Studies already exist where PhenoGraph performance has been tested using artificial two-dimensional data [184]. Given that PhenoGraph presents an alternative mathematical solution to the other software platforms investigated here, then it remains useful for users to see the characteristics of its clustering on a basic level compared with other methods, and be better informed about the choice of software solution for their specific data analysis task.

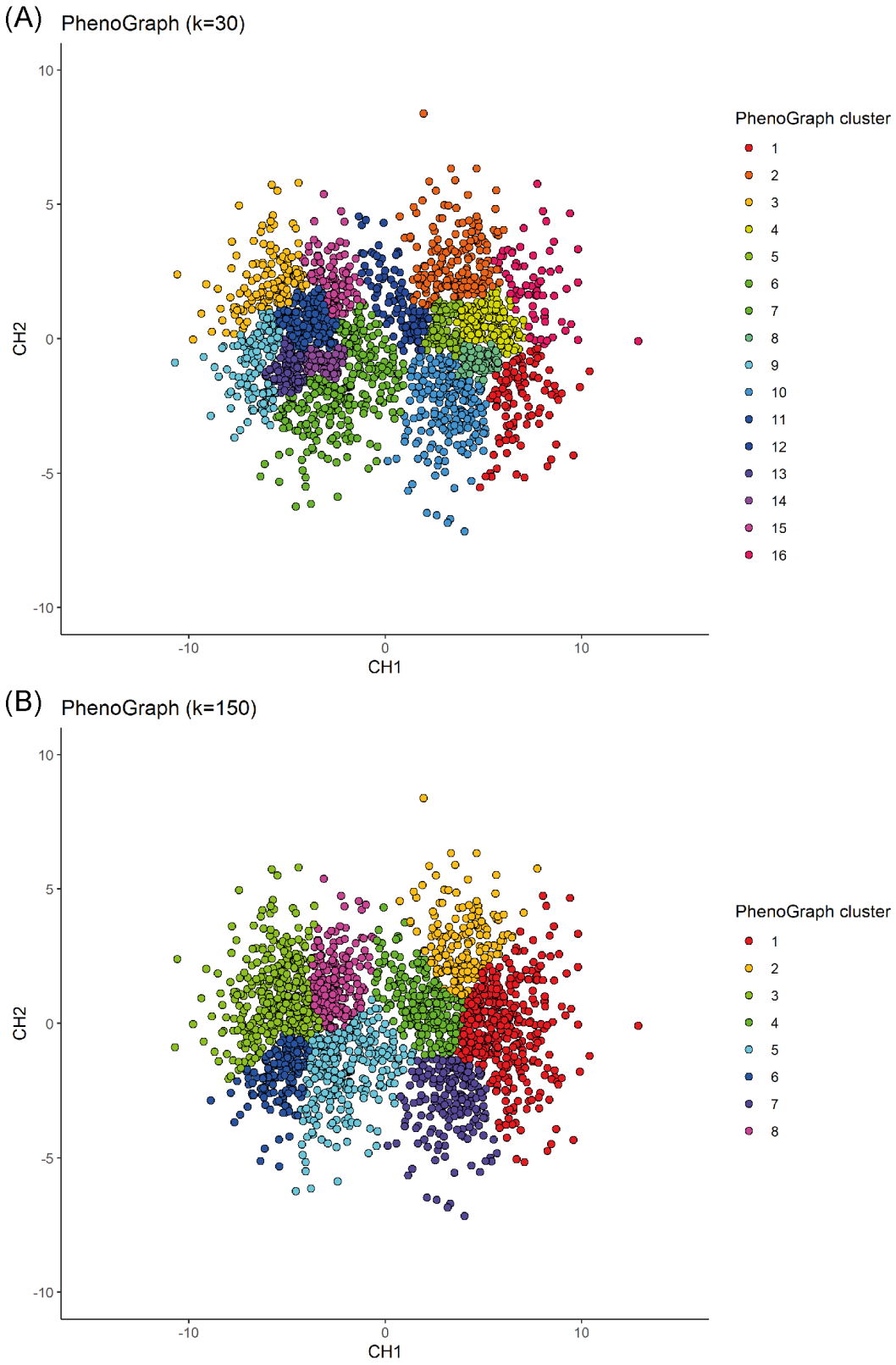


Figure 4.1: The effect of the input parameter k on the output number of clusters in PhenoGraph.

4.2.2.5 SPADE1

SPADE1 analysis was performed on the web-based Cytobank platform [119]. The SPADE1 runs on synthetic datasets could not be completed and analysis was not continued.

4.2.2.6 SPADE3

SPADE3 was run within Matlab R2019a. Each FCS file was run separately with no pooling (combining separately downsampled data into one meta-downsampled dataset), noting that all other platforms did not offer pooling. SPADE3 outputs were partitioned into two final populations using the semi-automated partitioning tool, with all suggested partitions being accepted. If three final populations were desired, the larger subpopulation from the first split would be selected for further partitioning.

4.2.2.7 SWIFT

SWIFT was run within Matlab R2019a. The SWIFT output number of clusters did not always match the input cluster number, and minimal manual interpretation was sometimes required to join separate clusters together to return the desired two or three clusters.

Table 4.2: User parameter settings for software runs.

Software	**Parameter**	**Setting**
Flock2	Bins	Auto
	Density	Auto
	Calculate centroids using	Mean fluorescence intensity
flowMeans	Max number of clusters	3 (values below 3 invalid)
FlowSOM	Number of expected metaclusters	3
	Grid size	3×3
PhenoGraph	k, number of nearest neighbours	150
SPADE1	Target number of nodes	Failed run
	Downsampled events target	Failed run
SPADE3	Outlier density	1st percentile (default)
	Target density	20,000 cells (default)
	Number of desired clusters	100 (default)
SWIFT	Input cluster number	2
	Arcsinh transformation	0

4.2.3 Statistics and performance evaluation metrics

The software outputs were recorded, and the absolute difference between cell populations of cluster 1 to the reference value was calculated in percentage terms, as in Eq. 4.1.

$$\text{Difference to reference } \% = \frac{|A - B|}{\text{Total events}} \times 100 \tag{4.1}$$

where A is the reference cluster 1 count, and B is the software cluster 1 count. The means of the difference to reference percentage values of each group were calculated, along with the sample SDs and CVs to assess repeatability.

Binary classification metrics have not been included in this Chapter because, as noted in Chapter 3 (Section 3.3.4), the values were objectively difficult to interpret in the absence of given acceptability thresholds, and less informative compared with statements of accuracy and repeatability expressed as the difference to reference count ±1SD.

4.3 Results

4.3.1 Output number of clusters

This study assessed the performance of the six automated data analysis software, each of which implement different clustering algorithms to identify and quantify cells: Flock2, flowMeans, FlowSOM, PhenoGraph, SPADE3, and SWIFT (density-based, k-means, self-organising map, k-nearest neighbour, deterministic k-means, and model-based clustering, respectively).

The initial investigation was focussed on whether the software could partition the datasets to give the same number of clusters originally designed into them. It was found that returning the desired number of clusters was straightforward for tools such as flowMeans, where the input number of clusters (k) directly determined the output. Obtaining the desired number of clusters from other software was more complex. In SWIFT, the input k served as an initial estimate which sometimes varied from the final output cluster number after subsequent cluster splitting and merging processing steps. In SPADE3, the default user settings automatically over-clustered the data into a minimum spanning tree with hundreds of nodes, with a subsequent 'semi-automated' feature to suggest tree partitioning to the user. Here, the tree partitioning step was applied until the desired number of clusters were produced. PhenoGraph, and occasionally Flock2 and SWIFT, tended to over-cluster the data, so additional manual steps were performed to merge sub-clusters together. Over-clustering is taken to mean the algorithm gives large

numbers of clusters that can be an order of magnitude greater than the desired number of clusters.

In general, the manual workload increased in proportion to the number of clusters generated by a software above the desired amount, illustrating a paradox of increased human intervention in a supposedly automated process designed to reduce operator variation. This study also found that flowMeans and FlowSOM did not permit outputs of two clusters, so processing of two-cluster datasets returned a minimum cluster number of three, thus again requiring a manual merging step.

Overall, strategies to obtain the desired output number of clusters varied significantly between different software, with some requiring repeated tuning of input parameters and/or post-clustering manual interpretation steps, suggesting a high level of operator training required, as opposed to casual use. This is an important facet when considering that the intention of automated software is often to reduce operator variation.

4.3.2 Clustering characteristics

The different software tools tested here all utilised different clustering algorithms, and certain data partitioning characteristics became particularly noticeable with overlapping clusters as the data because unstructured. Reference two- and three-cluster designs are depicted in Figures 4.2A and 4.3A, respectively, with SI values of -0.2, 0.0 and $+0.2$, along with the raw software clustering outputs, before manual intervention was performed to merge sub-clusters together from e.g., Flock2, flowMeans, FlowSOM and PhenoGraph. Scatterplots of the software clustering results show how neighbouring clusters from Flock2 and flowMeans were separated with hard straight line boundaries often radiating from a central region (Figures 4.2B, 4.2C and 4.3B, 4.3C). Whereas divisions among FlowSOM, PhenoGraph and SPADE3 clusters resembled meandering twisting lines that had echoes of underlying merged sub-clusters (Figures 4.2D, 4.2E, 4.2F and 4.3D, 4.3E, 4.3F). Clusters from SWIFT had softer boundaries, with the fitted Gaussian models visible that slightly overlap each other (Figures 4.2G, 4.3G).

4.3.3 Two-cluster separation

To assess the performance of software as cell populations come closer together, synthetic two-cluster datasets were generated with multiple replicates at each separation index condition (as described in Section 4.2.1.1).

While clusters remained separate and distinct with a SI ≥ 0.1, all software outputs were similar to the reference value (differences ranged from 0.01% to 0.97%), and strong

repeatability was observed (all standard deviations below 0.8%). However, as the two clusters came closer together and the SI approached and fell below 0.0, all six software platforms displayed a decrease in performance; the differences between the software values and the reference value widened, and repeatability deteriorated as demonstrated by the extent of the error bars (Figure 4.4). The critical SI region appeared to be around −0.1, any further overlapping of clusters resulted in sharp reductions in software performance and erratic outputs. To place this in the context of real data, the identification of chimeric antigen receptor (CAR)-T cells (e.g., on the basis of the CD19 protein) routinely requires the analysis of less well-separated clusters that fall into this SI region of −0.1 [185]. Overlapping clusters appeared to have the most detrimental effect on Flock2 performance, with differences to the reference value widening from (3.0 ± 4.1)% at SI −0.1 to (11.9 ± 9.6)% at SI −0.2. flowMeans showed similar trends of reduced performance, with difference to reference of (6.1 ± 4.0)% at SI −0.1 and (9.6 ± 4.3)% at SI −0.2 . In contrast, the smaller differences in SWIFT outputs to reference from (1.4 ± 0.73)% to (5.7 ± 2.6)% at SI −0.1 and −0.2 respectively indicated somewhat better detection of overlapping normally distributed cell populations. However, SWIFT was not able to return two clusters at SI −0.3.

Overall, application of the synthetic two-cluster separation dataset revealed that SWIFT performed better compared to FlowSOM, followed by SPADE3 and PhenoGraph in terms of accuracy and repeatability.

4.3.4 Three-cluster separation

Evaluation of the effect of cluster separation on software performance was extended by introducing another cluster to the dataset. The three-cluster dataset added an additional level of complexity because the software now had to make two partitions in the dataset rather than one. Having three clusters also negated issues such as FlowSOM giving a minimum cluster of three for the two-cluster dataset. After causing each software to return three clusters, the number of points per cluster was recorded and the population count of Cluster 1 was arbitrarily selected to compare against the reference cluster count of 1,000 out of 3,000 total events.

The results displayed similar trends in accuracy and repeatability to the two-cluster dataset (Figure 4.5). All of the software maintained good accuracy and repeatability at SI ≥ 0, with the exception of FlowSOM at SI 0.1, which displayed lower performance than others. As the SI decreased below 0, software performance again began to deteriorate. The reduction in performance for all software was again particularly noticeable from SI

-0.1 to -0.2. Below SI -0.2, the deterioration of performance appeared to plateau for flowMeans, PhenoGraph and SPADE3. Given that it showed consistently smaller differences to the reference value at SI ≥ 0 than other software, flowMeans appeared to be less affected by overlapping clusters, however whether this was a merit of the software or a consequence of 'random' equal partitioning of the dataset is questionable. Flock2 did not identify three clusters at a SI -0.2, and SWIFT at SI -0.3, further highlighting regions of the separation index dataset where clusters became difficult to resolve. Again it is noted that three-cluster partitioning is prevalent in manual cell analysis, for instance in the separation of blood cell populations: lymphocytes, monocytes and granulocytes on FSC vs SSC plots.

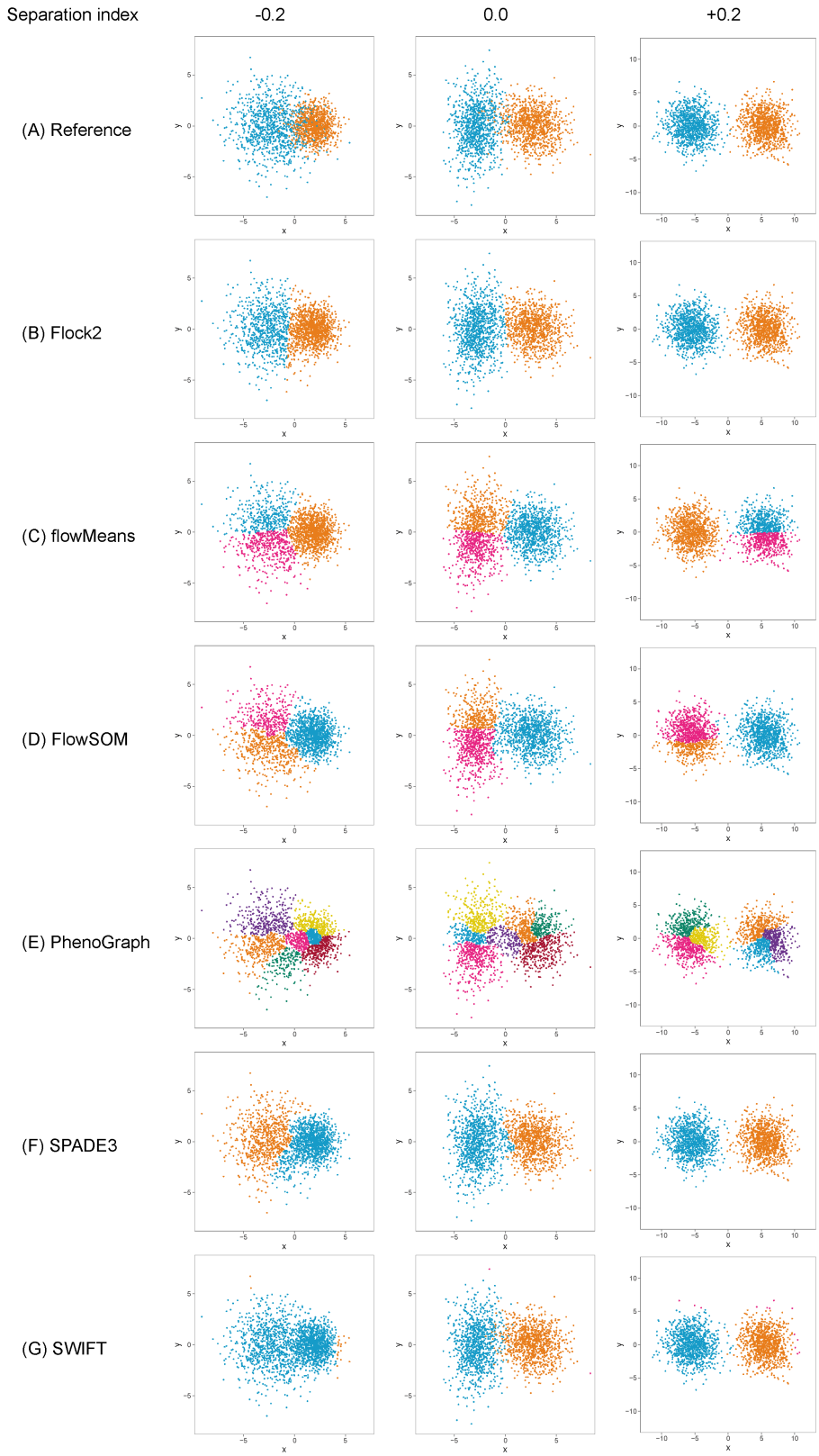


Figure 4.2: Clustering examples from different software on a two-cluster synthetic flow cytometry dataset with different degrees of separation between clusters.

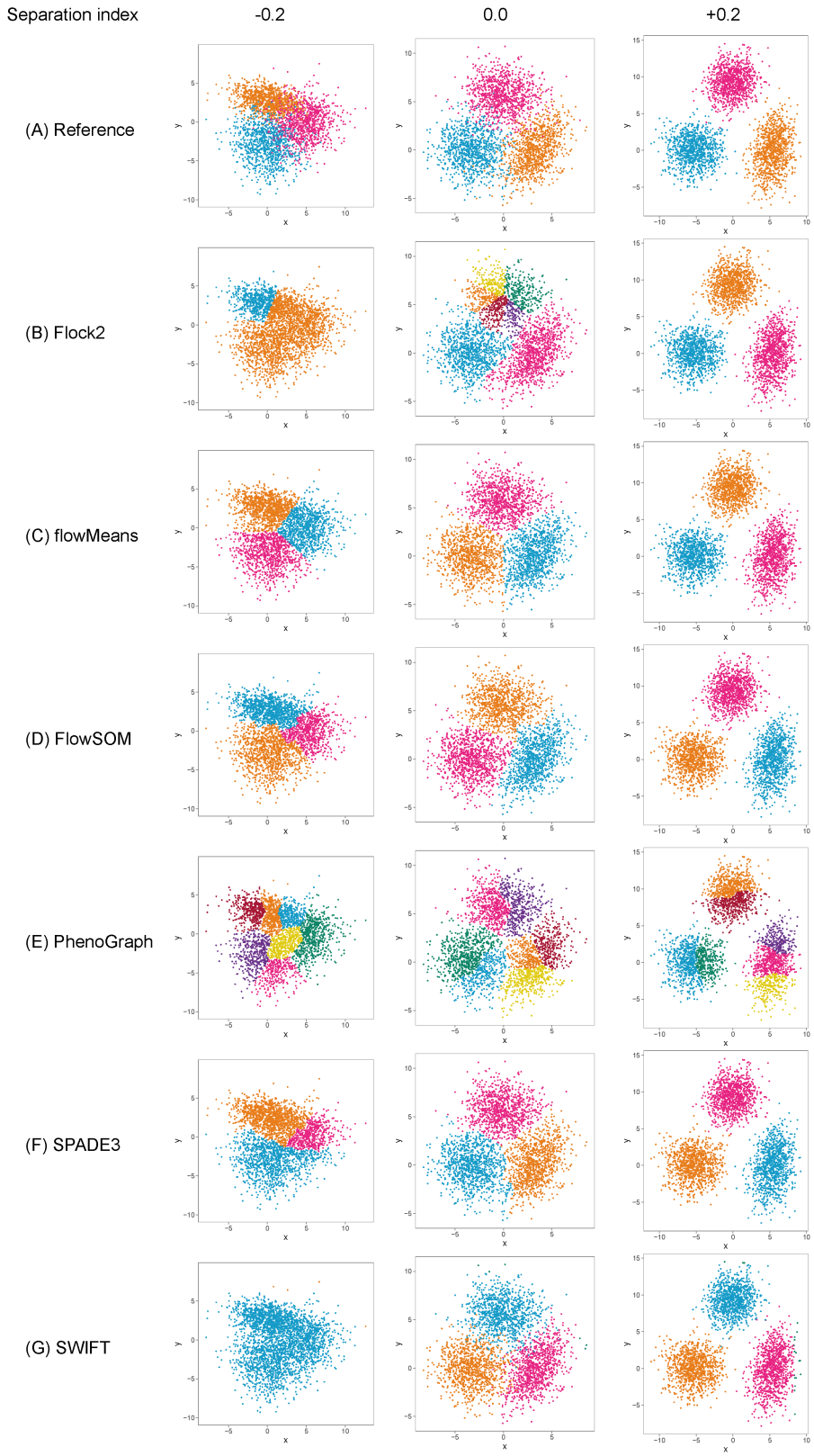


Figure 4.3: Clustering examples from different software on a three-cluster synthetic flow cytometry dataset with different degrees of separation between clusters.

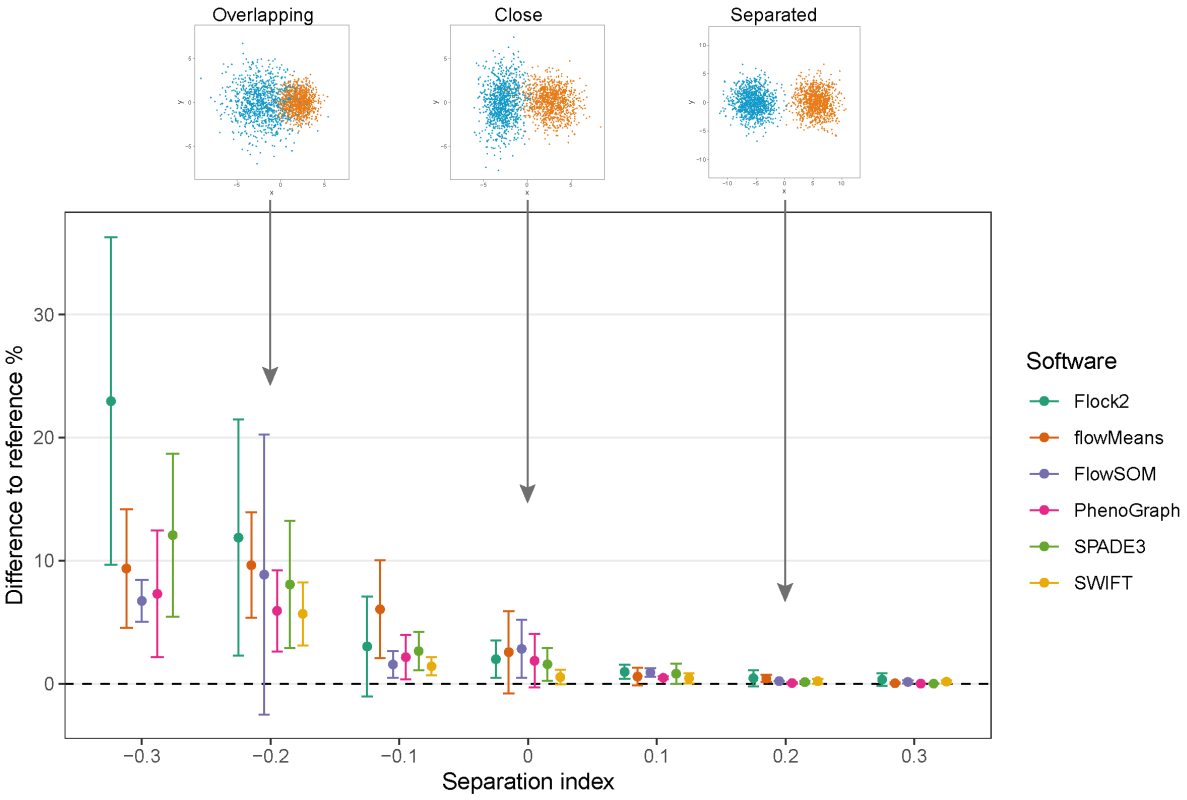


Figure 4.4: Performance of different software with a two-cluster separation dataset.

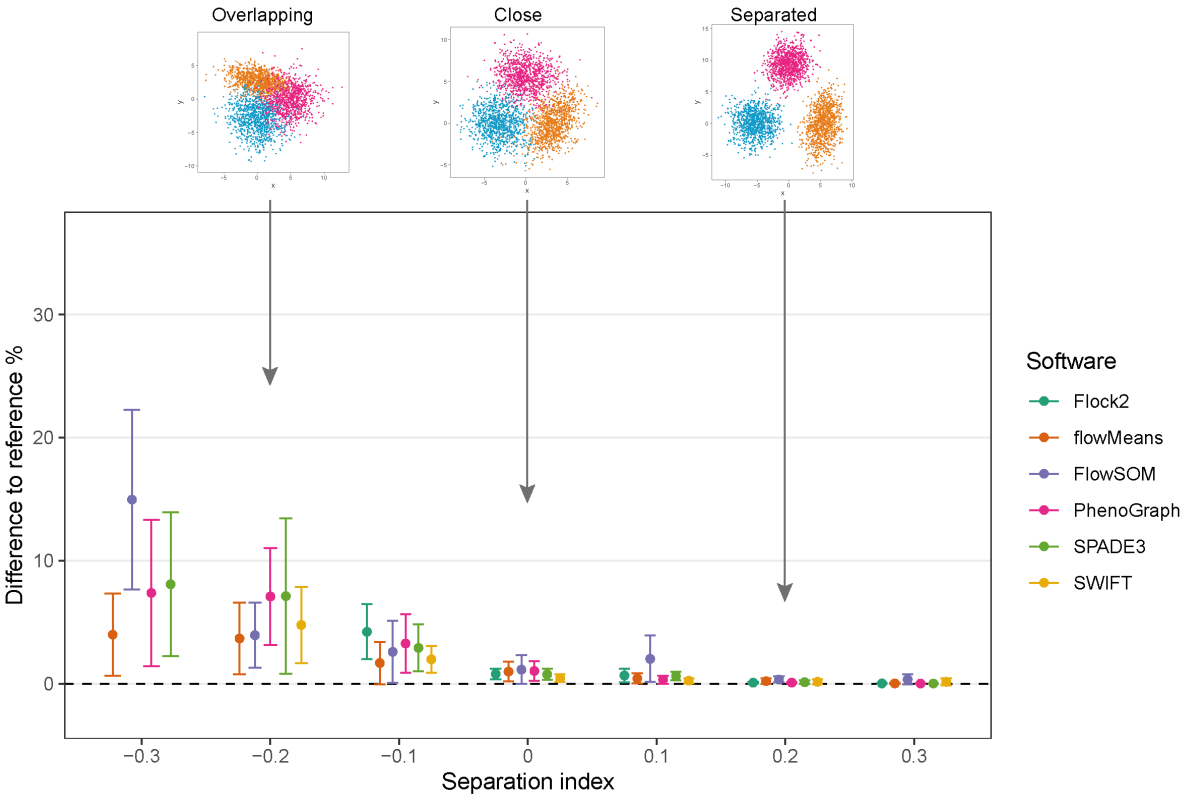


Figure 4.5: Performance of different software with a three-cluster separation dataset.

4.3.5 Skew

To understand whether the behaviours of software were limited to clusters with normal distributions, datasets containing clusters ranging from normal symmetrical to more asymmetrical skewed distributions were generated and processed through the software. Initial runs were performed on skew cluster pairs with a tail-to-tail orientation used here as an exemplar of real flow cytometry data.

The results showed that once again, different software returned different clustering outputs and partitioning characteristics from the same dataset (Figure 4.6). Obtaining the desired output number of clusters, two, was straightforward with Flock2, flowMeans, SPADE3 and SWIFT. FlowSOM gave a minimum output of three clusters, resulting in the appearance of a horizontal bisect of one of the two populations. PhenoGraph outputs partitioned the data into approximately eight clusters as a result of the k value that was selected as a compromise between excessive manual intervention and long computational run times (see Section 4.2.2.4).

Software responded to increasing levels of skew in different ways. In clusters with heavy skew, Flock2, FlowSOM and SPADE3 appeared to partition the data in a more similar manner to the reference dataset compared with flowMeans, PhenoGraph and SWIFT (Figure 4.6). In this tail-to-tail configuration, Flock2 outputs showed improved accuracy and repeatability as the levels of skew increased, going from a difference to reference of (23.5 ± 16.1)% at no skew (α= 0) to (4.2 ± 3.0)% at heavy skew (α= 10) (Figure 4.7). The opposite effect was observed for PhenoGraph, with the gap to reference widening from (5.2 ± 3.9)% at skew $\alpha = 0$ to (9.7 ± 8.4)% at skew $\alpha = 10$. In comparison, other software outputs showed no significant differences in performance as illustrated in Figure 4.6. A weak trend was observed for SPADE3 to have better accuracy and repeatability as the level of skew in the datasets increased, and the opposite trend (slight decrease in performance) was observed for SWIFT (Figure 4.7).

4.3.6 Skew orientation

It was thought that as well as the level of skewness, the orientation of skew clusters to each other could be a factor affecting a software's ability to identify cell populations. To investigate this further, the two-cluster skew dataset (initially orientated tail-to-tail), was extended to include cluster pairs facing both head-to-head and head-to-tail directions (Figure 4.8). Again it was seen that whilst most software were able to return two clusters, FlowSOM returned three clusters, and PhenoGraph overclustered the data.

The extension of the skew dataset revealed SWIFT to be the software most affected

by skew clusters. In the head-to-head configuration, the gap to reference declined from $(2.6 \pm 2.2)\%$ at skew $\alpha = 0$ to $(35.7 \pm 21.6)\%$ at skew $\alpha = 7.5$ (Figure 4.9). Furthermore, SWIFT failed to return any output at skew $\alpha = 10$. The head-to-head pairings also showed flowMeans decreased in performance with increasing skew, with difference to reference going from $(9.8 \pm 4.5)\%$ at skew $\alpha = 0$ to $(18.0 \pm 4.5)\%$ at skew $\alpha = 10$.

Comparison across all software suggested FlowSOM and SPADE3 were least affected by skew distributions, both outperformed Flock2 and flowMeans in terms of accuracy and repeatability.

In the head-to-tail orientation, SWIFT's performance was noticeably lower than other software at every level of skew above 0 (Figure 4.10). For instance, the difference to reference of $(21.3 \pm 3.0)\%$ at skew $\alpha = 7.5$ was worse than the average of all other software $(7.5 \pm 3.8)\%$. This suggested the strategy SWIFT utilises to fit data to Gaussian distributions followed by splitting and merging steps may be challenged by the processing of non-Gaussian distributions.

An alternative visualisation of the results from the skew dataset runs suggest most of the software tested showed a decline in accuracy and repeatability as the orientation shifted from tail-to-tail, to head-to-tail and then head-to-head, respectively (Figure 4.11). For instance, at skew $\alpha = 7.5$, the reduction in performance from tail-to-tail, to head-to-tail and then head-to-head was shown for Flock2 ($5.4\% \pm 5.4\%$, $7.5\% \pm 4.7\%$ and $16.8\% \pm 14.6\%$, respectively), FlowSOM ($4.3\% \pm 3.8\%$, $6.4\% \pm 3.9\%$ and $8.5\% \pm 6.6\%$, respectively) and SWIFT ($4.6\% \pm 3.7\%$, $21.3\% \pm 3.0\%$ and $35.7\% \pm 21.6\%$, respectively), to give a few examples. This pattern was generally observed at all levels of skew tested.

The changes in performance was likely due to the reduction in the density of events in between the two clusters moving between one orientation to the other, i.e. the higher density of interface events in the head-to-head orientation made data partitioning more difficult. An interesting exception to this pattern was observed with PhenoGraph, where analysis of tail-to-tail skew clusters appeared to slightly reduce in accuracy and repeatability compared with the head-to-head orientated skew clusters. This was possibly because of characteristics of the PhenoGraph algorithm, or more likely that the significant manual intervention required to merge output clusters together to achieve final outcomes artificially improved PhenoGraph results.

Taken together, automated analysis of these synthetic skewed datasets revealed the effects of skew on software performance were largely software dependent, and affected different classes of clustering algorithms in varying ways. Software that model Gaussian distributions onto data were the least well performing (flowMeans and SWIFT). Density-based clustering software appeared to be unaffected by skew characteristics in the data

(Flock2). FlowSOM, SPADE3 and PhenoGraph performed well against other software tested here, potentially because they implement overclustering steps that break up the data into smaller populations that each differ in skew properties from the main major population.

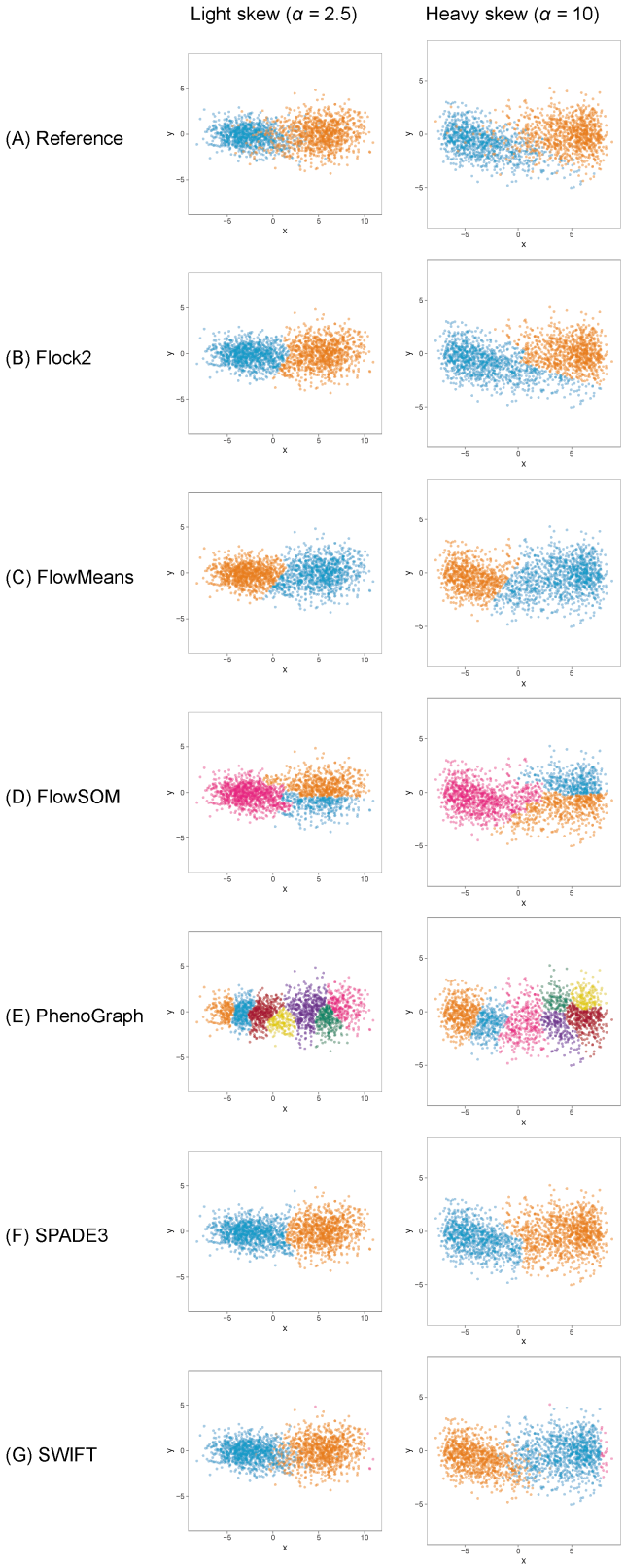


Figure 4.6: Clustering examples from different software on a two-cluster dataset with skew distributions. Two levels of skew are shown, light skew ($\alpha = 2.5$) and heavy skew ($\alpha = 10$), with cluster orientations all facing tail-to-tail.

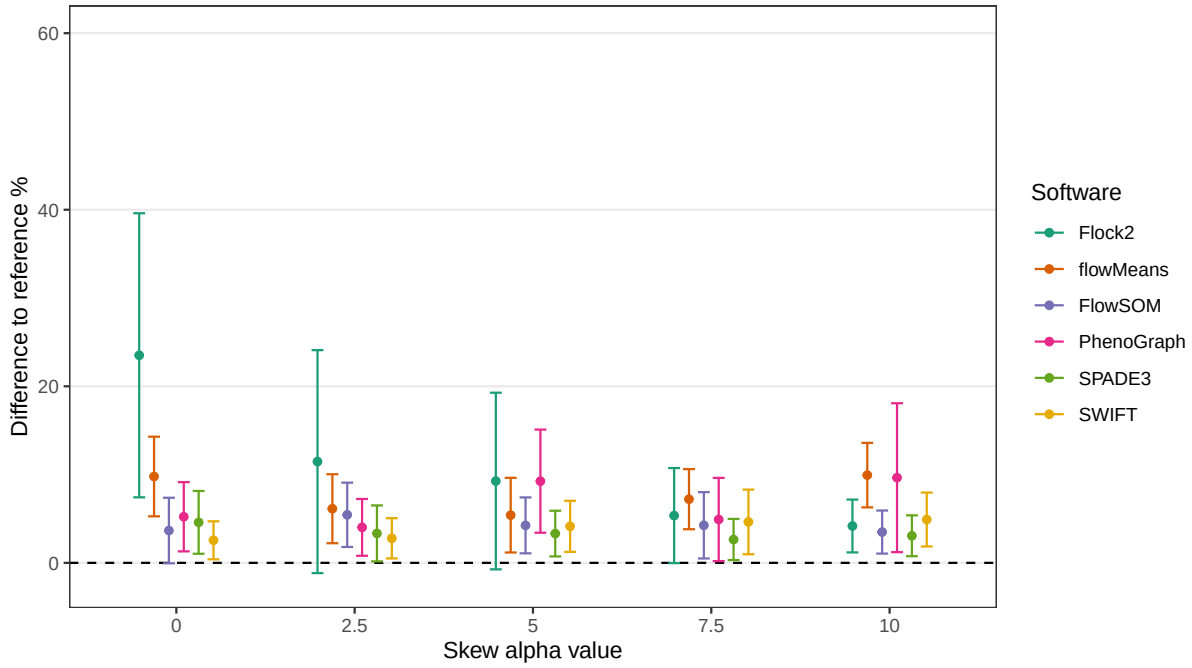


Figure 4.7: Performance of different software on a dataset with skew cluster orientations facing tail-to-tail.

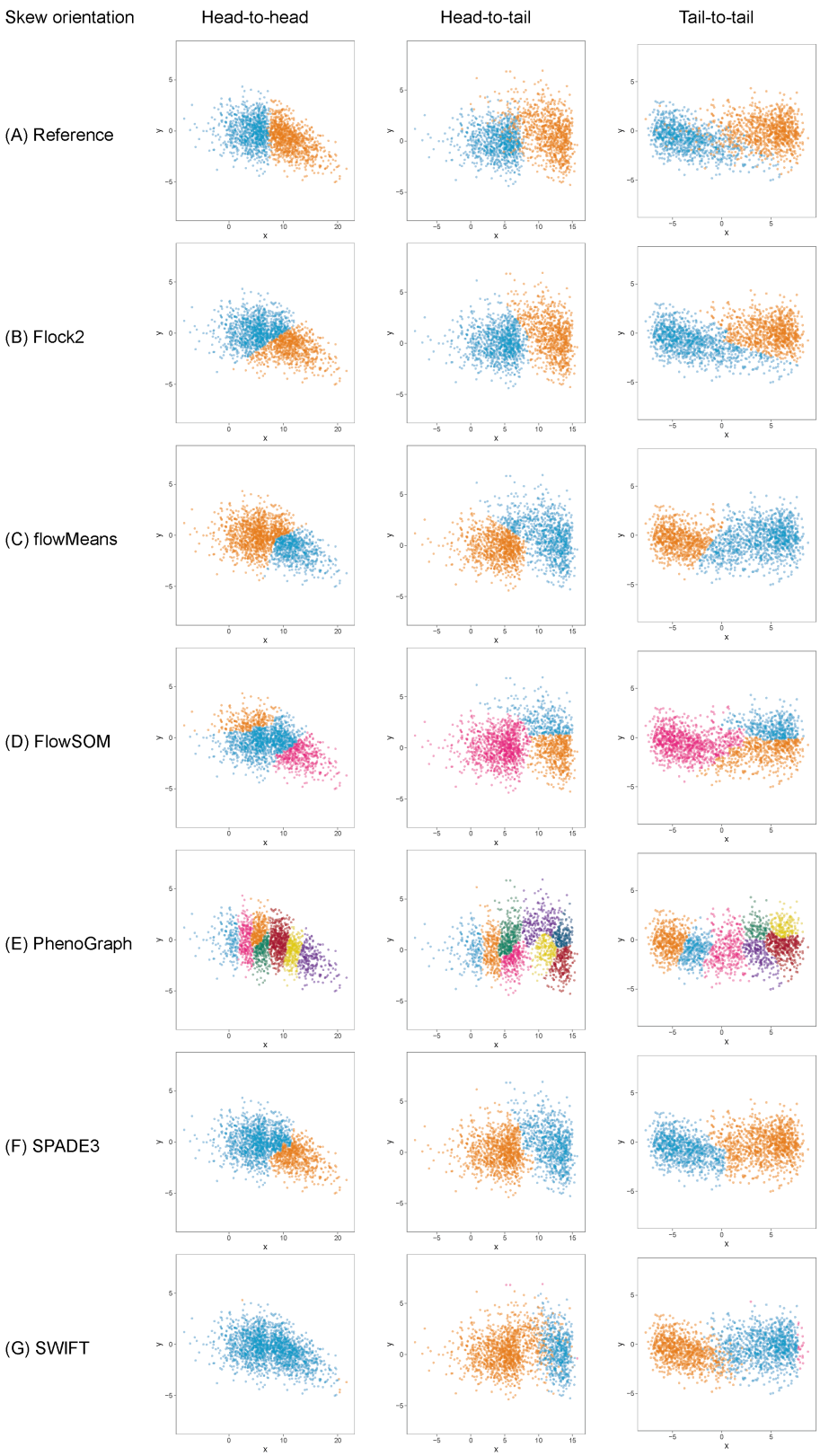


Figure 4.8: Clustering examples from different software on a two-cluster dataset with skew pairs facing different orientations. All clusters shown with heavy skew ($\alpha = 10$).

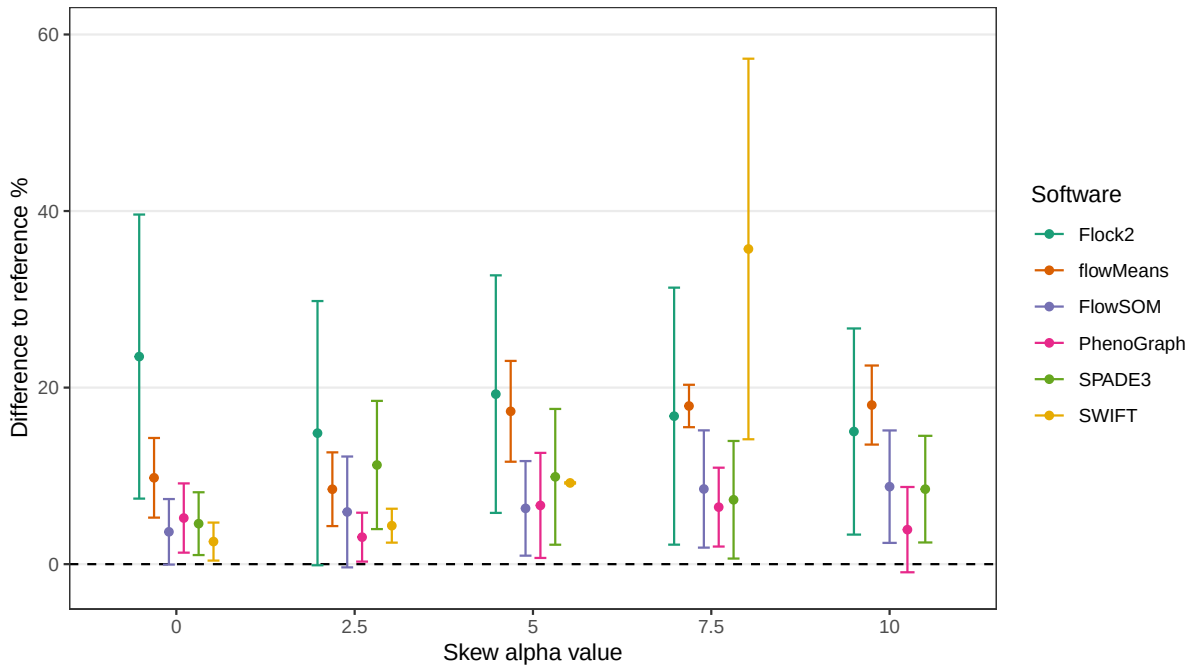


Figure 4.9: Performance of different software on a dataset with skew cluster orientations facing head-to-head.

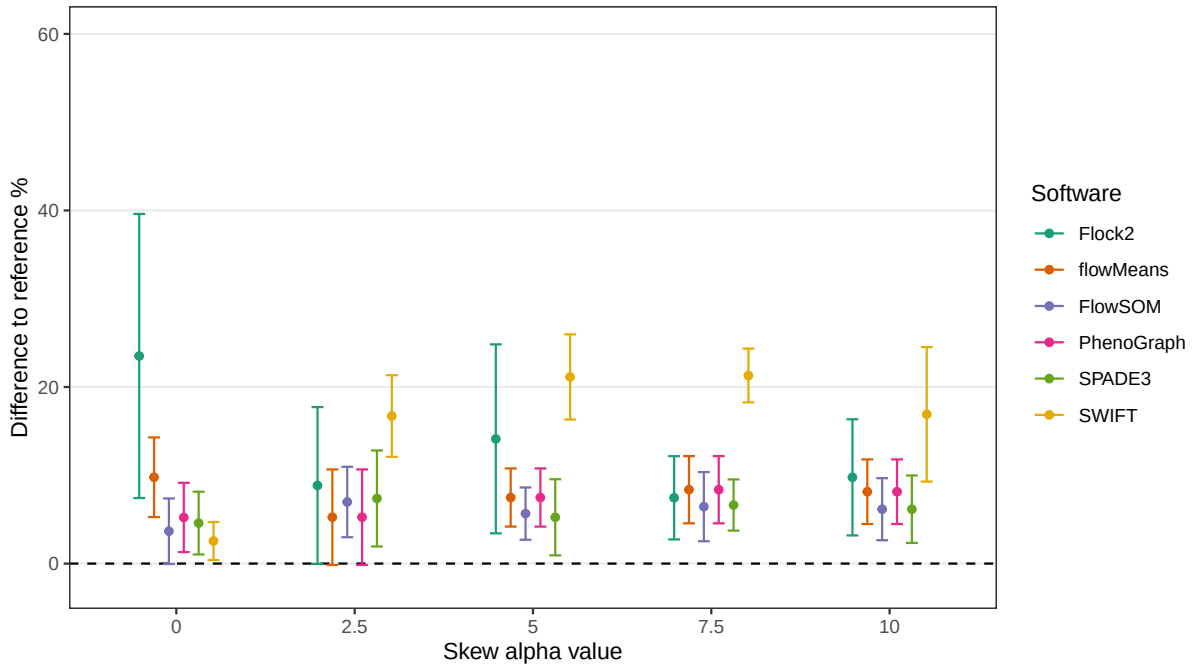


Figure 4.10: Performance of different software on a dataset with skew cluster orientations facing head-to-tail.

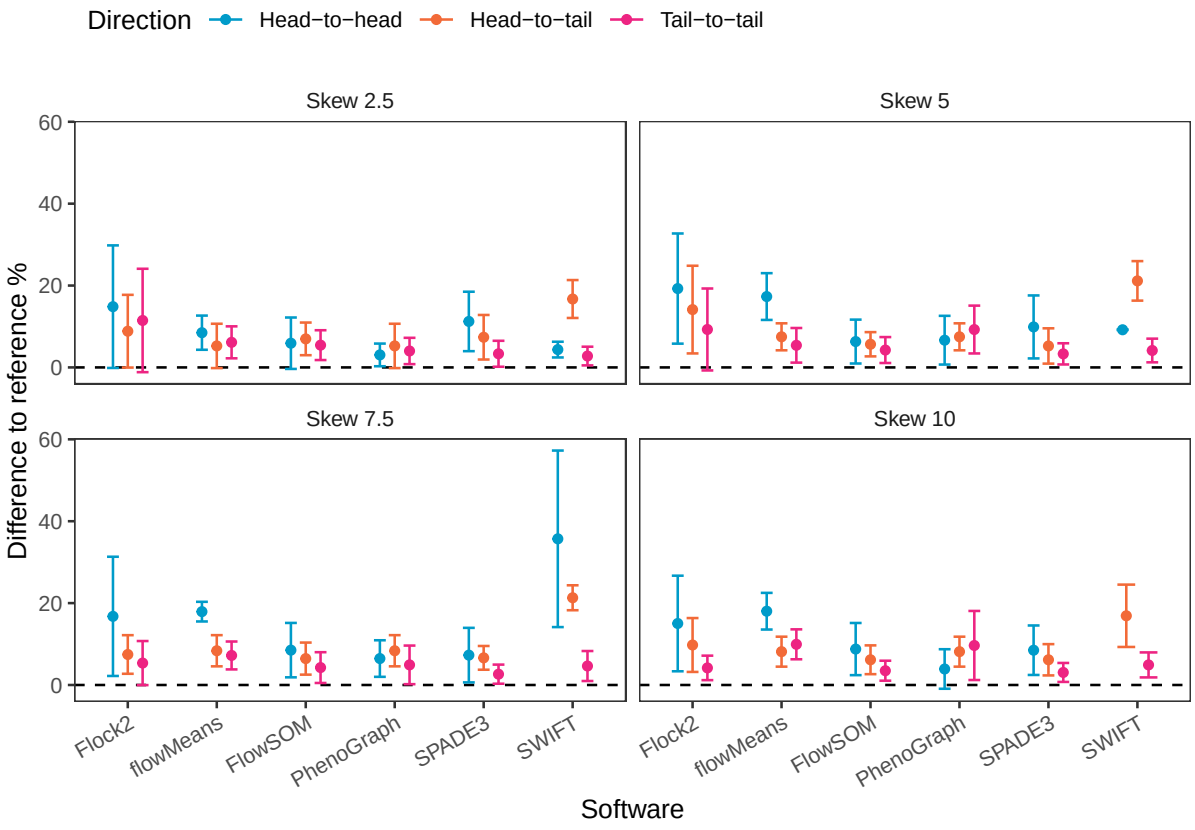


Figure 4.11: Performance of different software on a dataset with skew cluster orientations facing head-to-head, head-to-tail, and tail-to-tail.

4.4 Discussion

Characterisation of ATMPs by automated flow cytometry data analysis software have the potential to improve the quality, repeatability, and robustness of biomanufacturing processes by reducing operator variation as a function of subjective manual gating of clustered data. However, the lack of clarity in how these software derived outputs from inputs, coupled with the absence of toolsets for software validation and standardisation potentially restricts their implementation by the manufacturing community. In addition it presents challenges from a clinical and regulatory perspective.

Previous work on the inter-comparison between synthetic and a real dataset showed clear visual correlation among cell distribution characteristics examples (Chapter 3). Consequently, for this particular cross platform comparison there was confidence that the synthetic data mirrors the key characteristics of low dimensionality cluster data, demonstrates design flexibility and application, and allows for traceable benchmarking (absolute accuracy and repeatability), without the further need to run the platforms through further real data. Nevertheless, the results obtained with synthetic datasets here are further validated with actual experimental data containing cell populations with different distances between clusters and skewed data distributions later in this thesis, in Chapter 7.

In this study, synthetic datasets have been designed and applied to test the performance of six automated flow cytometry cell population identification computational tools. The use of synthetic datasets with controlled distances between clusters demonstrated similar patterns of behaviours between different software, in which accuracy and repeatability deteriorated as clusters came closer together, particularly below the separation index value of -0.1. These software responses were expected given that overlapping clusters change from multi-modal to unimodal distributions, progressively becoming one large cluster with merged cell populations. The skew datasets implemented here identified considerable variation in outputs between software when processing non-Gaussian distributed clusters, reflecting the different mathematical approaches employed by software to identify cell populations.

Among the six automated tools assessed here, the SWIFT algorithm was found to display better accuracy and repeatability compared to other tools as normally distributed clusters began to overlap and their separation index shifted below 0. However, when assessed further with skewed clusters, SWIFT performance noticeably declined more than others as the skew levels increased. Insights such as these can give operators unfamiliar with computational tools and algorithms a deeper understanding of the potential optimal working ranges of these tools, and the variations in performance that can arise between

them depending on the data structures. Furthermore, it could support upstream assay design to ensure data outputs are fit for automated analysis, such as switching to fluorophores leading to more optimised separation, or acquisition settings.

The synthetic dataset approach applied in this study to evaluate automated cell population identification tools extends on, but cannot be directly compared with findings from previous comparison studies, because of the differences in datasets (synthetic and real world) and dataset characteristics used. For example, studies have previously identified FlowSOM as high performing based on high dimensional datasets [47], however in this study SWIFT outperformed FlowSOM in the low dimensional, normally distributed dataset; although further testing in the presence of skewed clusters saw SWIFT performance deteriorate.

Compared with previous software comparison studies, the datasets applied here reduce the dependence on narrow cell model examples. Further strengths of this approach include the use of measurable distances between clusters through the separation index, as well as controllable skew parameters, with the benefit of allowing computational tools to be tested one-factor-at-a-time, on controlled sets of criteria not feasibly generated from experimental conditions. Of note, the synthetic datasets allow comparisons of software outputs away from subjective manually gated reference values that lack a 'ground truth' thus providing explicit statements of accuracy and repeatability.

This study specifically targeted the variation arising from data analysis within the flow cytometry analytical process. Upstream sources of variation from starting materials, sample processing, and instrumentation would require separate comparison studies designed around those factors as variables (e.g. conditions such as lysis, wash, staining) and with the data analysis software tool kept constant. With regards to the relevance of this data to biological samples analysis, the synthetic datasets here have been designed with essential properties that simulate their equivalent biological counterparts. Therefore, software runs that fail on encountering such data characteristics would directly infer on the (lack of) credibility of results from similar biological samples.

A recognised limitation of this work is that the number of markers simulated is lower than those in real data (usually $>$ 3-colour panels) because a priority in this study has been to understand and benchmark how algorithms behave with two or three clusters before introducing further complexities into the datasets. Noting the successful referencing and correlation study already completed between synthetic and real data (Chapter 3), overall, real data have been excluded from this initial Chapter because they are significantly more complex, containing sources of variation from upstream processes and noise components that cannot be controlled to transparently understand the 'black box' nature

of the algorithms investigated. Though this point is addressed later in Chapters 5 and 7 using clinically relevant data with rare cell populations, for inter-comparisons between synthetic datasets and real datasets to be performed.

Additionally, it is very difficult to achieve absolute cell counts for real data, so defining measurement accuracy (a critical component of this study) would not be possible. This research here has applied clearly defined synthetic datasets to establish the base functionality of software at lower numbers of parameters before escalating to higher dimensional datasets (i.e. we cannot run before we can walk). Having achieved this, building more complex datasets is the next key area for further work, and once at that stage, further comparisons between real datasets could illustrate even greater relevance.

The primary aims of the skewed datasets in this research were to investigate the first order effect of cluster distribution characteristics on software analyses, and currently, as a function of design of experiment, this has to assume a homogeneous cell population. To address the potential for a more heterogeneous cell population mix, further work could model cell subsets within the bulk-component of the skewed population with changing phenotypes (e.g., stem cells undergoing differentiation, T cells in response to cytokine activation), in line with the escalation of various components of complexity within the synthetic dataset design.

Previous work within this group has specifically quantified human operator variation when manually gating flow cytometry data that deals with the escalation of gating complexity, and the consequential deterioration of repeatability (hence increase in operator variation) [42, 43, 186]. Within this previous work there are clear indications of the threshold of processing or data discrimination ability of operators. This presents an interesting contrast of philosophy, because on the one hand manual gating is regarded as the "gold standard" for reference values within typical EQA schemes, yet automated software solutions are often regarded as a way of removing the operator variation, albeit this work has determined that many aspects of the six software platforms assessed herein require significant operator intervention to cause viable output of results. This suggests that the overhead of operator variation has potentially just been shifted elsewhere within some of the software environments. Without doubt this is another avenue of investigation for consideration in the future alongside the other complexity components discussed above, and also will (most probably) raise some significant regulatory questions about methodologies and potential redefinition of best practice or gold standards.

In this Chapter, the initial performance characteristics of six automated flow cytometry data analysis software have been evaluated using synthetic reference datasets. Overall, the results suggest that benchmarking of automated flow cytometry software platforms

will be possible with a high level of testing integrity using synthetic cluster datasets. The goal of this work was initially to enable biomanufacturers to make better informed decisions about whether or not to implement automated data analysis tools in their workflow instead of/ in addition to manual gating methods, based on their own cytometry data — although it is clear it is also relevant to the clinical community and would potentially impact regulatory science.

Where advanced analysis methods are deemed necessary, the clustering characteristics of different analysis tools illustrated here will facilitate the selection of ones that are fit for purpose. For users, these toolsets can be used to validate and verify installed software and confirm that working ranges match the specifications of their own data. For regulators inspecting process validation documentation, the inclusion of these datasets to provide assurances in automated cell characterisation measurement processes would potentially be desirable. There is potential for the development of synthetic digital reference materials to provide assurances in advanced analytical methods, leading to enhanced measurement confidence in ATMPs.

The results presented in this Chapter open up further work to explore more data properties in synthetic dataset design. In particular, rare cell populations is a subject for the work in Chapter 5, and noise parameters are assessed within Chapter 6.

4.5 Chapter conclusions

- Synthetic flow cytometry datasets containing specific distances between clusters and skewed population distributions were applied to different software tools.
- The two-cluster and three-cluster separation datasets both demonstrated a reduction in performance from all software as the SI fell below 0.
- Application of the skewed dataset showed software tools responded differently to increasing levels of skew, with results suggesting Flock2's accuracy and repeatability improved with heavier skew distributions, while PhenoGraph's worsened.
- The orientation of skewed clusters was thought to be a factor affecting software ability to identify cell populations, with the data suggesting a decline in accuracy and repeatability as the orientation changed from tail-to-tail, to head-to-tail, and head-to-head, respectively.
- Comparison across all software suggested SWIFT was most affected by skewed distributions, possibly because of its strategy to model Gaussian distributions onto the data.
- The work here has demonstrated the capability of synthetic datasets as benchmark-

ing toolsets for assessing the performance characteristics of software tools at their base functionalities.

Chapter 5

Rare cells detection

5.1 Introduction

Previous chapters have investigated the effects of two properties within flow cytometry data — distances between clusters, and skewed distributions — on the performances of software. So far, cluster sizes designed in these synthetic datasets have been equal. However, cell subpopulations in real flow cytometry data do not usually occur in equal frequencies. In fact, cell subsets that occur in low frequencies, such as rare stem cells, or circulating tumour cells, are often of greater biological significance. This chapter explores the variation in software performance when a rare cell population is introduced.

5.1.1 Definition and examples of rare cell populations

Definitions of what constitutes a rare cell population are variable in the literature depending on cell types, but generally refer to a frequency of 0.01% and below, i.e. 1 in 10^4 events [187].

Haematopoietic stem cells (HSC) that express the CD34 cell surface antigen are considered rare populations in peripheral blood and occur in frequencies of 0.01% – 0.1% [39]. These rare stem cells are biologically significant because they have the capacity to self-renew, and their daughter cells have the potential to differentiate into a variety of blood cells [188, 189]. Due to this multipotent ability, CD34+ stem cells are used in autologous and allogeneic transplantation for haematopoietic reconstitution following high-dose chemotherapy for cancer patients [190]. Flow cytometric quantification of these CD34+ cell populations is therefore an important process at all stages of HSC transplantation from initial harvest to graft assessment, to inform clinical care.

In a regulatory context, measurement of minimal residual disease (MRD) tumour pop-

ulations in haematological malignancies is used as a surrogate endpoint in clinical trials to demonstrate efficacy for drug treatments. The definitions of MRD are disease specific, but for example, guidelines from the US Food and Drug Administration (FDA) for acute lymphoblastic leukaemia (ALL), acute promyelocytic leukaemia (APL), and chronic lymphocytic leukaemia (CLL) accept an MRD level of less that 0.01% as supporting evidence of efficacy for progression-free survival in patients. Furthermore, the FDA recommends for the analytical sensitivity of the MRD assay to be at least 10-fold below the clinical decision-making threshold (i.e. detection level of 1 in 10^5 if MRD negative is defined as less than 1 in 10^4) [191]. For comparison, guidelines from the European Medicines Agency (EMA) define undetectable MRD, specifically for multiple myeloma (MM), as less that 1 in 10^5 residual tumour cells in the bone marrow following treatment [192].

Other examples of rare cell types include antigen specific T cells occurring at a frequency of 1 in 10^6, and rare tumour cells at 1 in 10^7 events in peripheral blood [193].

In the context of cell and gene therapy manufacturing, rare cells of interest may include populations that have a potential link to improved patient outcome, such as T memory stem cell populations in chimeric antigen receptor (CAR) T cells [194]. Alternatively, contaminating cell subsets present at low populations in culture, such as residual stem cells that have the potential to undergo uncontrolled differentiation when administered, could adversely affect the safety and efficacy of the cell therapy product. These rare cells would require careful in-process monitoring and characterisation.

5.1.2 Limits of detection and quantification

In rare event analysis, establishing the limits of detection (LoD) and limits of quantification (LoQ) of an assay are critical for measurement assurance. Definitions for determining the LoD and LoQ are variable in the literature, and can also be confused with terms such as sensitivity, analytical sensitivity and detection limit.

Guidelines from the Clinical and Laboratory Standards Institute (CLSI) define the following [195]:

- The background, or limit of blank (LoB), is defined as the highest apparent signal expected in the absence of the analyte of interest, calculated through replicate blank measurements (typically n=20) that estimate the mean and standard deviation (SD) as follows:

$$\text{LoB} = \text{mean}_{\text{blank}} + 1.645(\text{SD}_{\text{blank}}) \tag{5.1}$$

- The LoD is defined as the lowest analyte concentration reliable distinguished from

the LoB and at which detection is feasible, calculated as:

$$\text{LoD} = \text{LoB} + 1.645(\text{SD}_{\text{low concentration sample}}) \tag{5.2}$$

- The LoQ is defined as the lowest concentration of analyte that can be reliably detected and meet predetermined targets for bias, imprecision, and total error. The LoQ cannot be lower than the LoD.

Accuracy is defined in ISO 5725 as the closeness of agreement between the arithmetic mean of a large number of test results and the true or accepted reference value [196]. Since reference materials with absolute true values do not exist for cells, accuracy cannot be determined in flow cytometry. Alternative methods such as comparisons with other testing methodologies or proficiency testing programs to enable inter-laboratory comparisons can be used instead.

The threshold for precision, as expressed by the CV, differs from one cell or disease type to another and is largely driven by clinical need. Guidelines from the International Council for Standardization of Haematology (ICSH) and International Clinical Cytometry Society (ICCS) state that a CV of less than 10% is a desirable target for flow cytometry assay imprecision, but a more generous CV of 20% may be acceptable for rarer populations (occurring at frequencies of 1:1,000 or lower) [197].

The detection of rare cells by flow cytometry can be approximated with Poisson statistics, where events occur randomly and independently in a certain time period or volume with a constant rate. The effect of Poisson statistics means that increasing the number of counts of the cells of interest (rather than the total cells) increases the precision of analysis [198, 199]. Technically, to reach a given precision, the total number of rare events that need to be acquired, r, can be determined by the calculation:

$$r = \left(\frac{100}{\text{CV}}\right)^2 \tag{5.3}$$

and the total number of events required for statistically relevant analysis can be calculated based on the frequency ratio to the rare population of interest:

$$\text{Cell frequency} = \frac{\text{Rare events}}{\text{Total events}} \tag{5.4}$$

As an example, for a rare cell occurring at a frequency of 1 in 10^5, achieving a CV of 10% would require 100 target events acquired in 10^7 total events, whereas a CV of 20% would require 25 events acquired in 2.5×10^6 total events. Reference tables for determining the database/sample size needed for a given precision are available [193].

In flow cytometry, the LoD and LoQ are expressed in terms of the number of events. Consensus guidelines on MRD analysis recommends 30 rare cells as the minimum number of acquired events necessary to give an LoD of 1 in 10^5 (requiring 3×10^6 total cells), and 50 rare cells the minimum threshold for an LoQ of 1 in 10^5 (requiring 5×10^6 total cells) [200].

While these statistical considerations are important for data acquisition aspects of flow cytometry experiments, the data analysis (gating) process preformed downstream remain unchanged from non-rare population analysis. Data are visualised in sequential 2D dot plots and gates are drawn to exclude doublets, dead cells, and debris, and to isolate populations of interest. To guide the correct manual placement of gates on positive populations, fluorescent minus one (FMO) controls are used [170]. Thus, the variability associated with operators persists and the task has the potential to be automated.

For the unsupervised clustering algorithms that are the focus of this thesis, the problem with rare events in data lies in discriminating whether the group of data points in space is a cluster, but doing so without the aid of FMOs to determine how best to split the data (although automated gating pipelines that use pre-defined FMO gating templates have been developed [201]). A number of factors can influence the detection of rare cells by software tools, such as subsampling steps, clustering strategies, and the different statistical distributions used to model the data.

The concepts of LoD and LoQ for automated analysis of flow cytometry data have yet to be firmly established, along with the minimum number of cells in a cluster required for automated gating tools to identify a population with desired precision. To that end, there is scope to utilise synthetic datasets with controlled rare populations in comparison studies, in efforts towards understanding the potential LoDs of software tools.

5.1.3 Chapter aims

This chapter explores the confidence of automated data analysis software in rare population detection. The work here applies to both synthetic and real-world datasets that contain a rare cell population through various automated cell population identification tools. The results from this chapter are important for understanding the behaviour of different algorithms at low cell population counts, and what limitations there are to the techniques. This understanding will help provide guidance to users in diagnostic settings and in manufacture of cell therapy products.

The aims of this chapter are to:

- Generate synthetic datasets featuring rare cell populations, with normal and non-normally distributed clusters.
- Evaluate flow cytometry automated data analysis software performance for rare population detection, using synthetic datasets.
- Perform this evaluation on a range of software that utilise different clustering algorithms.
- Determine a range of metrics including accuracy, repeatability, and limits of detection of rare cells.
- Validate the software performance results carried out using synthetic datasets, using real cell datasets.
- Compare the clustering outputs of software, and rank their overall performances.

5.2 Methods

5.2.1 Synthetic datasets

5.2.1.1 Rare-normal dataset

The purpose of these datasets was to evaluate software performance in the detection of cell populations with increasing rarity. Since the occurrence of rare cell events is relative to the total event size, two-cluster datasets were prepared with total events of 10^3, 10^4, 10^5, and 10^6. The minimum size considered for a rare cluster was 10 events, increasing to 50, 100, 500, and finally 1,000 events the maximum. The dataset design is outlined in Table 5.1.

Clusters were generated using R package *clusterGeneration*. The clusters were normally distributed, well-separated, with a separation index value of 0.2. Three replicates of each condition were created. The files were converted to flow cytometry FCS 3.1 format using the R package *flowCore*.

5.2.1.2 Rare-skew dataset

The two-cluster rare dataset was extended to include clusters with non-normal skew distributions. Clusters were designed as before, with total events of 10^4, 10^5, and 10^6, and with size of rare cluster as 10, 50, 100, 500, and 1,000 events. Both clusters in the dataset were designed with heavy skew (alpha = 10); cluster pairs faced a head-to-tail orientation as an exemplar of real flow cytometry data. Skew clusters were generated as previously described in Chapter 3 using the R package *sn*. Three replicates of each condition were

Table 5.1: Synthetic rare dataset design

Dataset	Total events	Rare events	Rare %
1	10^3	10	1
2	10^3	50	5
3	10^3	100	10
4	10^4	10	0.1
5	10^4	50	0.5
6	10^4	100	1
7	10^4	500	5
8	10^4	1000	10
9	10^5	10	0.01
10	10^5	50	0.05
11	10^5	100	0.1
12	10^5	500	0.5
13	10^5	1000	1
14	10^6	10	0.001
15	10^6	50	0.005
16	10^6	100	0.01
17	10^6	500	0.05
18	10^6	1000	0.1

created.

5.2.2 Real cell datasets

Real cell datasets containing rare populations were sourced and also tailor-made to validate the software results obtained from synthetic datasets. Characteristics for the rare cell, real-world datasets were designed to match those from the synthetic datasets as much as possible, in particular a large dataset of up to 10^6 events and a rare cell frequency of 10 in 10^5 (0.01%) (Table 5.2).

5.2.2.1 CD34 dataset

The CD34 dataset used was pre-existing and the Author did not generate the biological material themselves. All material was obtained with the approval of the Ethics Committee of Loughborough University under the jurisdiction of the Human Tissue Authority. Cryopreserved mobilised peripheral blood CD34+ cells (Axol Bioscience) were thawed and expanded for six days in CD34+ expansion medium of IMDM (Gibco) supplemented

Table 5.2: Real-world rare cell dataset design specifications

Property	**Specification**	**CD34 dataset**	**K562 dataset**
Events per file	Up to 10^6	10^3	10^6
Rare cell frequency	10 in 10^5 (0.01%)	3 in 2.2×10^3 (0.1%)	67 in 8.9×10^5 (0.008%)
Number of subsets	Ideally 2 or 3, up to 4	8 (theoretical)	2
Parameters	FSC, SSC plus up to 3 fluorescent channels[a]	FSC, SSC, FITC, PE, APC	FSC, SSC, GFP
Separation	Well-separated	Well-separated	Well-separated
Distribution	Normal distribution with minimal skew[b]	Slight skew distribution	Slight skew distribution
Replicates	3	3	3

[a] Any more will be unused in analysis. [b] Noting this property cannot be easily controlled.

with 20% BIT 9500 (Stemcell Technologies), 100 ng/mL each of SCF and FLT-3L and 50 ng/mL of TPO (all Peprotech), then the medium was replaced with neutrophil expansion medium of Stemline II (Sigma) supplemented with 100 ng/mL each of TPO, SCF and GCSF (all Peprotech) over days 7-21. Cells were stained with antibodies CD34-PE, CD133-APC, and Lin1-FITC (CD3, CD14, CD16, CD19, CD20, CD56 cocktail, to gate out any mature/lineage committed cells) (all BD Bioscience). Cells were analysed on a FACSCantoII cytometer (BD) equipped with 3 lasers (405nm/30mW, 488nm/20mW, 633nm/17mW). Files acquired at each timepoint contained 10^4 events, which were subsequently split into thirds to $2.6 \times 10^3 \pm 3.2 \times 10^2$ total events in order to simulate 'triplicates' to test repeatability of software (the original dataset lacked replicates). Manual gating of the data found that samples from day 15 onwards were candidates for rare cell populations, therefore conditions from days 15, 18 and 21 were taken forward for automated data analysis.

5.2.2.2 K562 dataset

The K562 dataset was generated by Dr Shiqiu Xiong at the National Measurement Laboratory hosted at LGC. K-562 and K-562-GFP cells (both ATCC) were cultured strictly following manufacturer's protocols [202, 203]. 1×10^8 K-562 cells at exponential growth phase were harvested, washed twice in PBS (Gibco), resuspended in 5 mL and split into 5 tubes labelled A-E, 1 mL per tube. 4×10^7 K-562-GFP cells in exponential growth

phase were harvested, washed twice in PBS and distributed into 5 tubes by titration from 2×10^7 to 2×10^3 sequentially, with each tube made up to 1 mL with PBS. Each K-562-GFP cell sample was mixed with K-562 cells in A-E tubes, to make K-562-GFP:K-562 ratio from 1:1 to 1:10,000. Samples were acquired using a BD LSRFortessa cell analyser equipped with four lasers (355nm/20mW, 405nm/50mW, 488nm/50mW, 640nm/40mW). Samples were run in triplicate; 1×10^6 K-562 events were acquired per file. The 1:1,000 and 1:10,000 conditions containing suitable 'spiked-in' rare GFP+ cell populations were taken forward for automated data analysis.

5.2.2.3 Data processing

The major lymphocyte populations from both CD34 and K562 cells datasets was gated on FSC-A vs SSC-A plots using the autogating tool based on probability contours in FlowJo V10. To aid manual analysis, visualisation and interpretation of data, fluorescent channel data were rescaled using the logicle transformation function in the R package *flowCore*.

5.2.3 Software runs

The input parameters used for software runs are listed in Table 5.3.

5.2.3.1 Flock2 analysis

Flock2 analysis was performed on the web-based platform ImmPort Galaxy version 1.2 [182]. FCS files were uploaded to the platform and converted to a text file using the 'Convert FCS to Text' tool.

5.2.3.2 FlowSOM analysis

FlowSOM analysis was performed on the web-based platform ImmPort Galaxy version 1.2 [182]. FlowSOM (Galaxy Version 1.0) was run using the user parameters listed in Table 5.3. It was found that, rather than the value given for number of expected metaclusters, it was the grid sizes that specified the output number of clusters, e.g. a grid size of 3×3 forced three clusters to be returned. A grid size of 2×2 was an invalid input; two clusters could not be returned.

5.2.3.3 PhenoGraph analysis

PhenoGraph analysis was performed on the R platform using *Rphenograph* version 0.99.1 [183], with the user parameters listed in Table 5.3. The larger datasets required a two-

Table 5.3: User parameters for software runs on rare datasets

		Dataset		
Software	**Parameter**	**Synthetic**	**CD34**	**K562**
Flock2	Bins	30 (max)	30 (max)	30 (max)
	Density	2 (min)	2 (min)	2 (min)
	Calculate centroids using	Mean fluorescence intensity	Mean fluorescence intensity	Mean fluorescence intensity
FlowSOM	Number of expected metaclusters	3	8	3
	Grid size	3×3	10×10	10×10
PhenoGraph	*k*, initial clustering	30	120	30
	k, meta-clustering	15	N/A	15
SPADE1	Target number of nodes	Failed run	8	3
	Downsampled events target	Failed run	10%	10%
SPADE3	Outlier density	1st percentile (default)	1st percentile	1st percentile
	Target density	100,000 cells	20,000 cells (default)	20,000 cells (default)
	Number of desired clusters	100 (default)	100	100
SWIFT	Input cluster number	2	2	2
	Arcsinh transformation	0	0	0

step PhenoGraph processing strategy because the initial output number of clusters were impractical for minimal manual interpretation (> 8). PhenoGraph meta-clustering was performed on medians of each marker within each initial cluster (based on an approach used in [204]). The rare meta-cluster was then identified through a final manual interpretation step.

5.2.3.4 SPADE1 analysis

SPADE1 analysis was run on the Cytobank web-based platform. The platform failed to analyse synthetic datasets, so only runs from real cell datasets were successfully completed. SPADE1 was run on real cell datasets using the user parameters listed in Table 5.3.

5.2.3.5 SPADE3 analysis

SPADE3 analysis was run within Matlab R2019a. Datasets were processed with no compensation, no transformation, and user input parameters as listed in Table 5.3. Each FCS file was run separately with no pooling, noting that all other platforms did not offer pooling. SPADE3 outputs were partitioned into two final populations using the semi-automated partitioning tool, with all suggested partitions being accepted.

5.2.3.6 SWIFT analysis

SWIFT analysis was run within Matlab R2019a, with user input parameters as listed in Table 5.3. The SWIFT output number of clusters did not always match the input cluster number, and minimal manual interpretation was sometimes required.

5.2.3.7 flowMeans analysis

flowMeans analysis was not completed for the rare cells datasets because of issues with the plugin on the FlowJo platform that failed to deliver results.

5.2.4 Statistics and performance metrics

Statistics and performance metrics were performed as described in Chapter 3 (section 3.2.10). For the rare cell analysis in this chapter, a further binary classification metric was used in addition to those described in Section 3.2.10:

$$\text{Specificity} = \frac{\text{True negative}}{\text{True negative} + \text{False positive}} \tag{5.5}$$

Reference values were taken either from the known truth in synthetic datasets, or from cell counts from manual gates performed by an operator in real cell datasets. Software rankings were ordered based on the mean value of metrics averaged across all runs in a dataset.

5.3 Results

The datasets used in this study all include a rare cell population from 10% down to 0.001%. Since the rare cell frequency is determined by the rare cell count relative to the total events in the dataset, datasets with different sizes were used. For the synthetic datasets, total events sizes of 10^3, 10^4, 10^5, and 10^6 were used. For real cell datasets, sizes ranged from approximately 10^3 to 10^6 total events.

5.3.1 Results of rare-normal dataset runs

The synthetic datasets with clusters of normal distributions was first processed through several automated data analysis software to test their rare cell detection performances. The clustering outputs from software were manually interpreted, and the number of events (within clusters containing known rare events) were counted. Software performances were expected to decrease as rare clusters became smaller.

5.3.1.1 Total events 10^3

The synthetic dataset of 10^3 total events were processed through the different software, and the clustering outputs compared against the reference dataset are depicted in Figure 5.1. Outputs that (incorrectly) partitioned the larger population into subsets with rough, unstructured borders were observed for Flock2, FlowSOM, PhenoGraph and SPADE3. FlowSOM routinely gave three cluster outputs due to algorithm limitations (as explained in 5.2.3.2) and manual identification of the rare cluster output was required — this was true for all dataset runs performed in this study. PhenoGraph also gave a higher number of clusters than desired for the majority of its runs, and required substantial manual intervention to identify the rare cluster among its outputs (as explained in 5.2.3.3).

At 10 rare events (1% rare frequency), SWIFT appeared to be the only software able to identify the rare cluster, reporting counts of 6.3 ± 1.2. At 50 rare events (5% rare frequency), the rare population was detected by Flock2, FlowSOM and SWIFT (counts of 41.3 ± 15.9, 48.3 ± 0.6, and 46.3 ± 2.3, respectively). These three software were then

able to detect the increasingly large rare population of 100 cells (10% rare frequency). PhenoGraph and SPADE3 were unable to detect the 100-cell cluster with high accuracy or repeatability, as indicated by their reported counts of 443 ± 162 and 200 ± 174.

5.3.1.2 Total events 10^4

The software clustering outputs compared against the reference dataset are illustrated in Figure 5.2. Partitions that split the dataset approximately midway, occasionally with meandering lines, were observed for Flock2, FlowSOM, and SPADE3. For the rarest conditions these partitions were based on the larger cluster, ignoring the rare events. PhenoGraph characteristically appeared to partition the dataset with twisting branches that radiate away from the densest region of the dataset, i.e. the centre of the major cell population (Figure 5.2D). FlowSOM and PhenoGraph gave outputs that required manual intervention, as mentioned previously above. SWIFT appeared to disregard any events located near the limits of the dataset (including the rare ones) that did not fit a Gaussian distribution (Figure 5.2F).

At the 10^4 dataset level, none of the software were able to accurately detect and isolate the 10-cell rare cluster (0.1% rare frequency)(Figure 5.3, top row). Although FlowSOM was able to identify the 10 rare events as part of a separate cluster to the non-target cells, this cluster also included a number of false positive events. At 50 rare events (0.5% rare frequency), differences in software performances became apparent. The 50 rare cells were accurately detected by Flock2 (counts of 50.0 ± 1.0), FlowSOM (49.3 ± 2.1), and SWIFT (48.0 ± 0). These three software were then able to detect further large clusters of 100, 500 and 1000 events. SPADE3 remained unable to detect the rare population until it reached 500 events (5% rare frequency), however even at this condition, 1 out of the 3 runs failed (counts reported: 495, 494, and 5,332), indicating low repeatability. PhenoGraph failed to accurately detect the rare population in all conditions tested.

Note that the main clusters from different levels of rare cell counts have slightly different elliptical shapes and orientations because of the variable covariance matrices used during cluster design and generation in order to simulate a range of cell populations from flow cytometry data (see Chapter 3). These changing shapes do not impact the cluster rarity, separation between the two clusters nor their normal distributions.

5.3.1.3 Total events 10^5

The rare dataset was extended by increasing the total events by one order of magnitude, to 10^5. Similar characteristics of the data partitioning by the software were seen as with

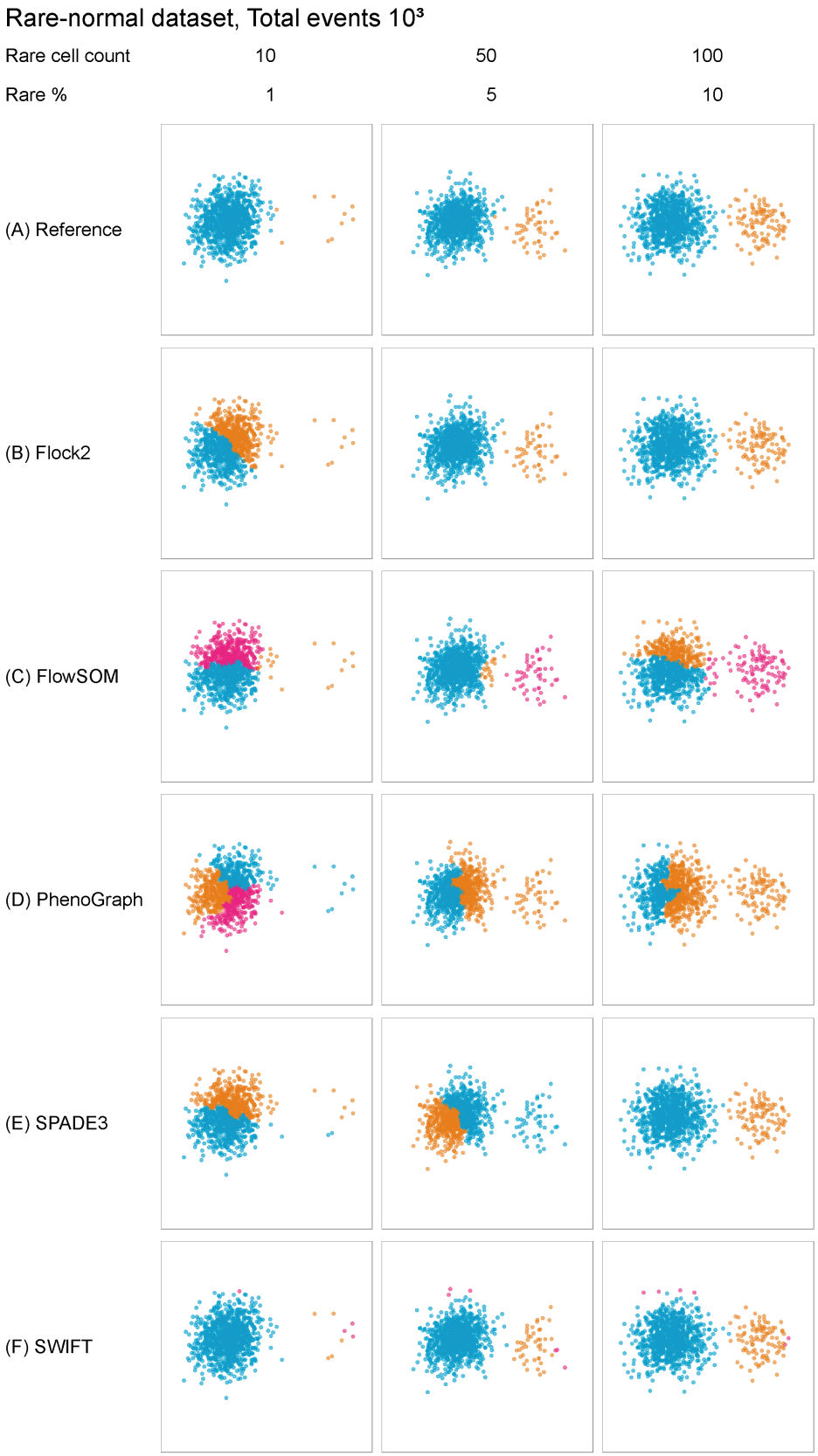


Figure 5.1: Clustering examples from different software on a synthetic two-cluster rare cells dataset, containing 10^3 total events, with normally distributed clusters.

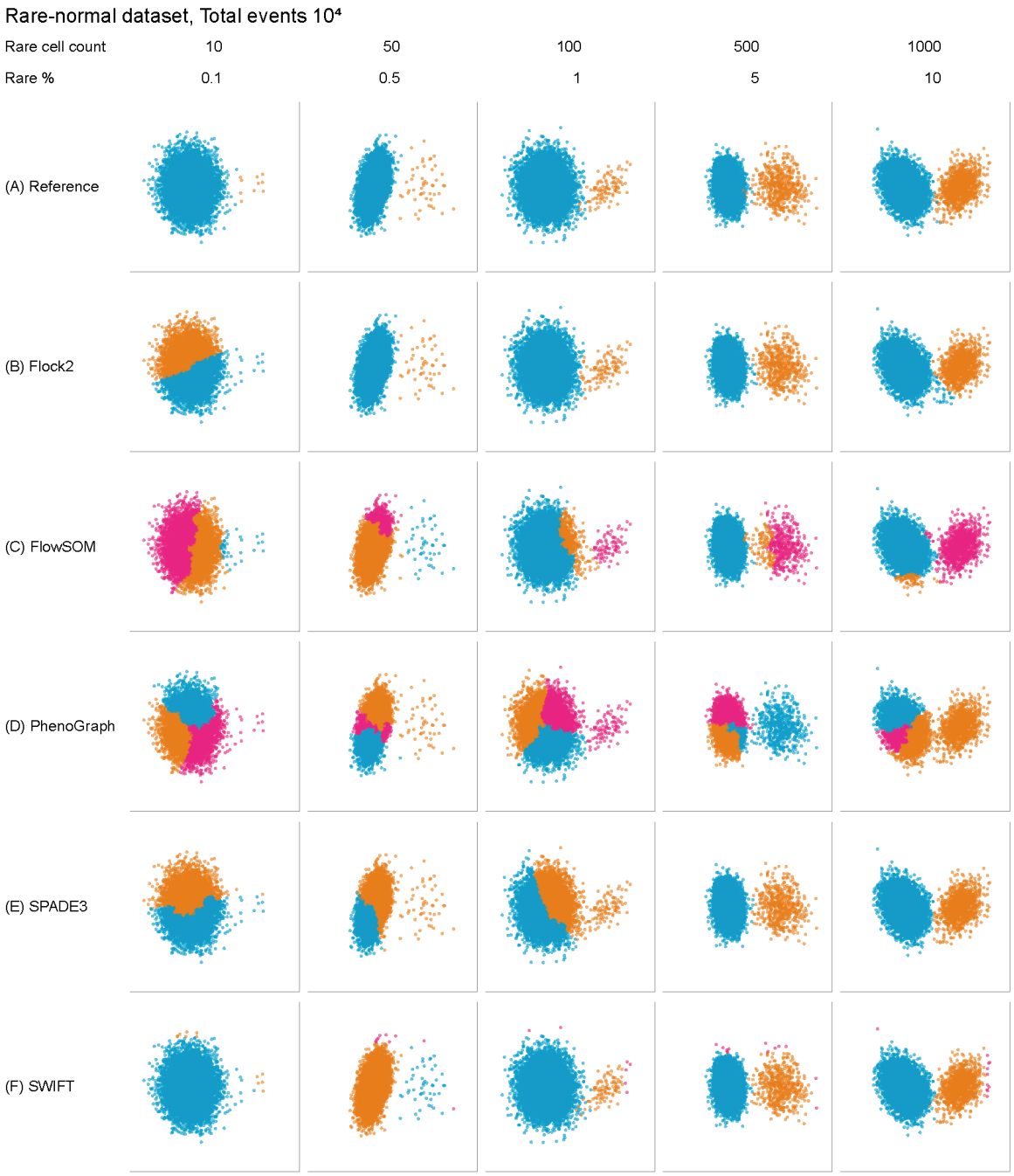


Figure 5.2: Clustering examples from different software on a synthetic two-cluster rare cells dataset, containing 10^4 total events, with normally distributed clusters.

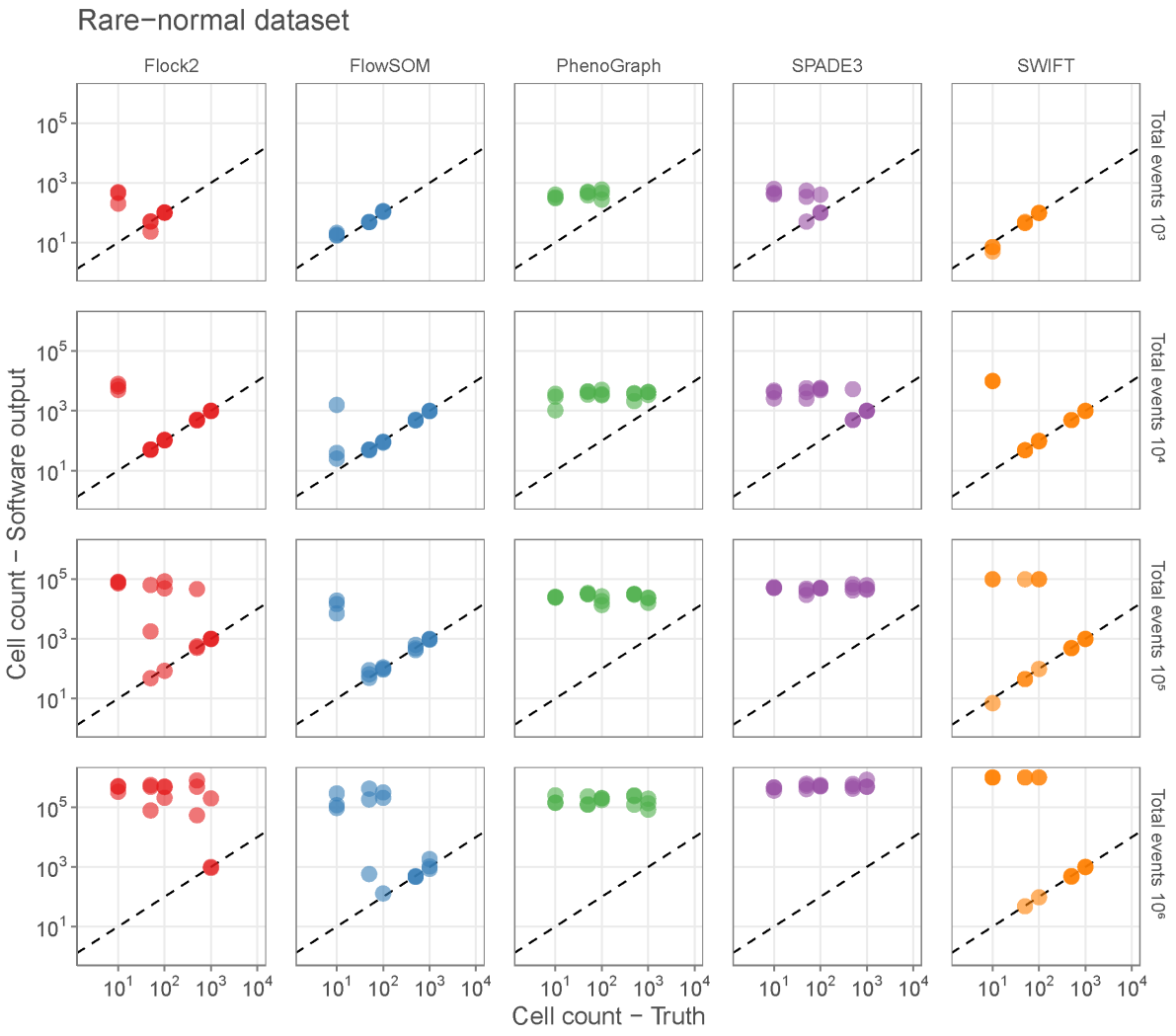


Figure 5.3: Plots of rare cell count truth vs. software output for each software (columns) and at different levels of total events (rows). Points on the dashed line indicate a 'hit'.

the 10^4 dataset (Figure 5.4). In most cases here, Flock2 and SPADE3 seemed unable to partition the data along the correct channel (x-axis) and appeared to split the data along a random orientation.

At 10 in 10^5 rare events (now 0.01% rare frequency), none of the software were able to detect the rare cluster with high accuracy and repeatability (Figure 5.3, middle row). At 50 in 10^5 events (0.05%), FlowSOM came closest to identifying the rare cluster, giving counts of 68.0 ± 20.7. At 500 events (0.5% rare frequency), SWIFT began to be able to detect the rare cluster (490.7 ± 4.6). Having failed to detect the rare clusters accurately at lower the sizes, Flock2 was finally able to detect 10^3 in 10^5 events (1%) with a count of $1,000 \pm 12$. However, rare cluster detection at this level still eluded PhenoGraph and SPADE3.

5.3.1.4 Total events 10^6

At total events of 10^6, detection of the rarer clusters should have been the most challenging test for the software. The comparison of clustering outputs against reference dataset (Figure 5.5) revealed mainly random partitioning by most software, apart from SWIFT, which did not partition the data where the rare cluster appeared to fit the Gaussian distribution of the major cluster, after discarding events on the limits.

Following the results from the 10^4 and 10^5 datasets, here, in addition to none of the software being able to detect the rare cluster of 10, the rare clusters of 50 (0.005%) and 100 (0.01%) events were also undetected (Figure 5.3, bottom row). At 500 events (0.05%), FlowSOM and SWIFT were able to detect the rare cluster with good accuracy and repeatability, reporting counts of 476 ± 17 and 484 ± 17 respectively. Flock2 was able to detect the rare cluster at 1,000 events (0.1%) in only 2 out of 3 runs, demonstrating limited repeatability. Similar to the 10^5 dataset, PhenoGraph and SPADE3 failed to detect any of the rare clusters at this level.

5.3.1.5 Performance analysis

Overall, for all levels of total events, no software was able to detect the rare cluster containing 10 cells, with the exception of SWIFT in the 10^3 total events condition. The limits of detection (LoD) varied considerably between software and the different levels of total events, although there was a slight trend for the LoD to improve in percentage terms as the total events increased (Figure 5.3). SWIFT and FlowSOM appeared to show stronger performance, with LoDs improving from 0.5% to 0.05% going from the datasets containing 10^4 to 10^6 total events. Flock2 reached a LoD of 50 cells in 10^4 total events,

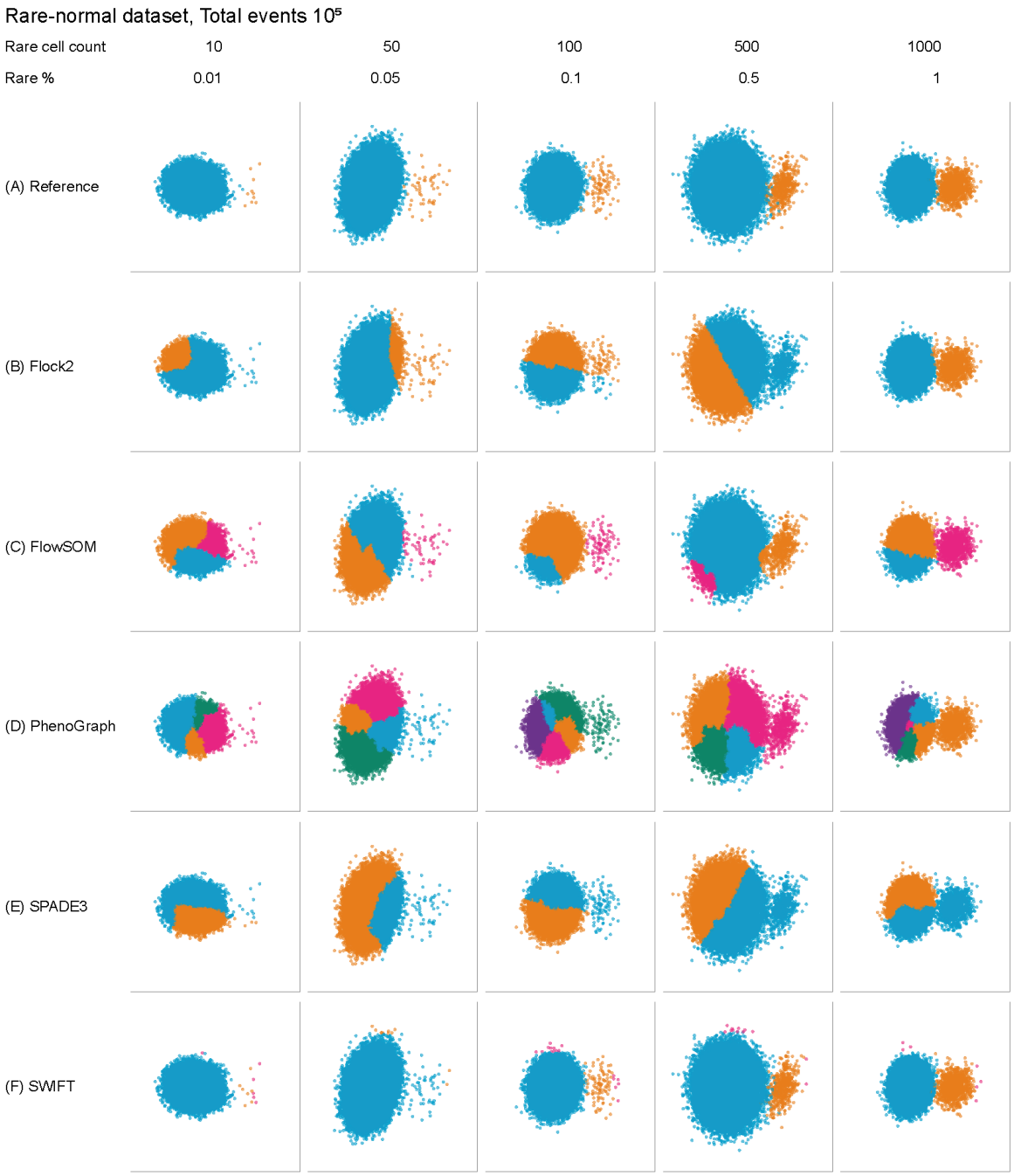


Figure 5.4: Clustering examples from different software on a synthetic two-cluster rare cells dataset, containing 10^5 total events, with normally distributed clusters.

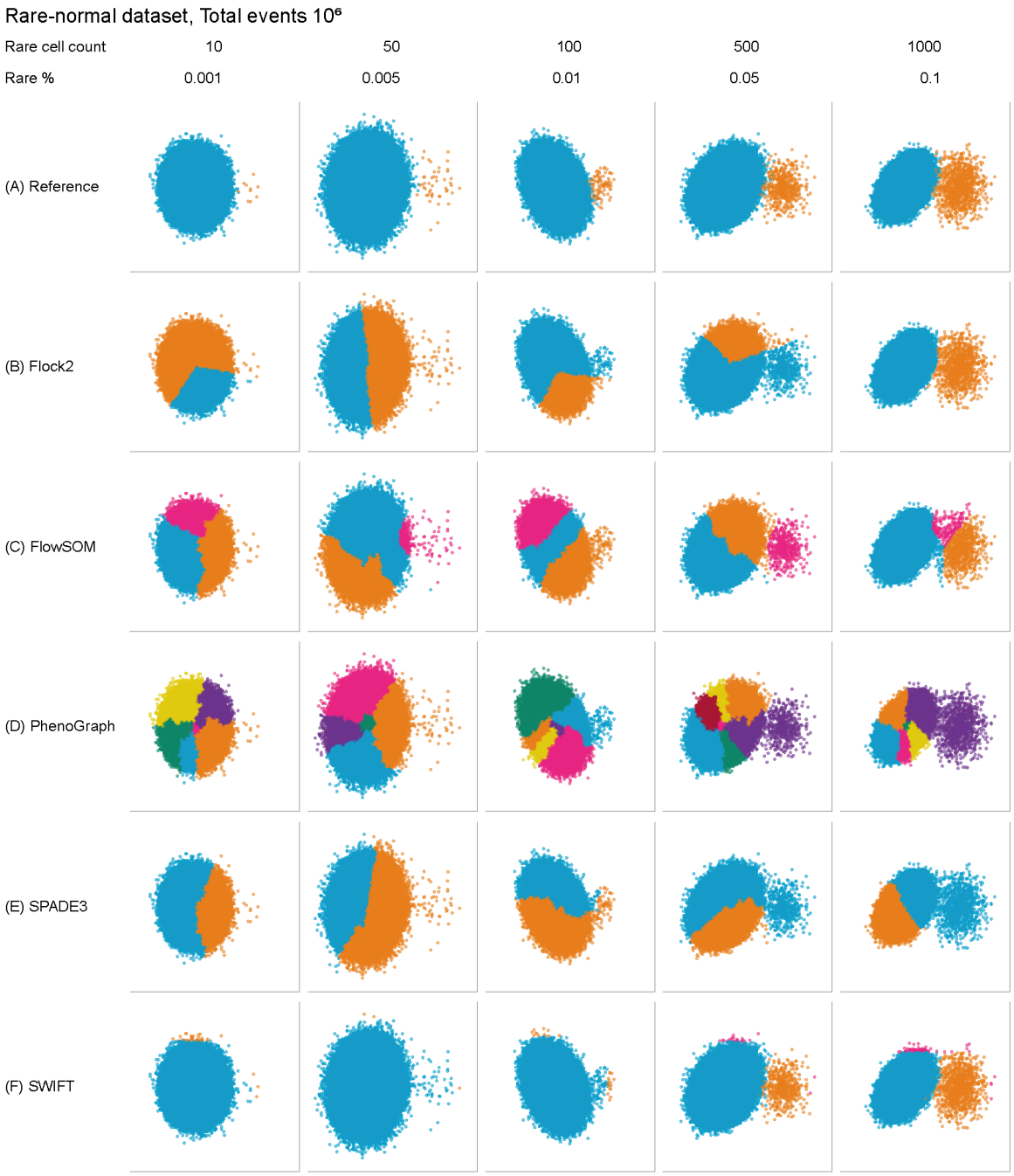


Figure 5.5: Clustering examples from different software on a synthetic two-cluster rare cells dataset, containing 10^6 total events, with normally distributed clusters.

and its performance deteriorated as the total number of events increased. PhenoGraph and SPADE3 failed to detect the rare population as the total number of events in the dataset increased.

The decision on which software showed reliable detection of rare cells will depend upon user predefined thresholds for accuracy and precision, which will clearly vary depending on context. As an example here, a detection threshold of the true rare cell count ±20% was used to enable a subjective interpretation of software performances rather than one based on operator judgement (Figure 5.6). Noting that different thresholds (e.g. at ±10%) will give different outcomes. Using this exemplar threshold, FlowSOM and SWIFT appeared to be the best performing software for detection of the rare cluster at total events of 10^3, 10^4, 10^5 and 10^6. Flock2 ranked below SWIFT, followed by SPADE3, and finally PhenoGraph was ranked lowest having failed all tests for detection of the rare cluster.

5.3.2 Results of rare-skew dataset runs

The rare cell detection performance of software was further investigated using a synthetic two-cluster dataset that contained skewed clusters. Clustering characteristics for the rare-skew datasets were similar to the rare-normal ones, at all levels of total events. In conditions where the rare cluster was more difficult to detect, Flock2, FlowSOM, PhenoGraph and SPADE3 split the main skewed cluster into fragments with meandering boundaries (Figures 5.7, 5.8 and 5.9). FlowSOM gave three cluster outputs that required manual interpretation (as explained previously in Section 5.2.3.2) (Figures 5.7C, 5.8C and 5.9C). PhenoGraph gave cluster outputs that increased in number as the total events in the dataset increased, and which required considerable manual intervention to isolate the rare cluster (as explained previously in Section 5.2.3.3) (Figures 5.7D, 5.8D and 5.9D). The splitting of the main cluster in SPADE3 outputs (and also PhenoGraph, to a certain extent) appeared to have shifted away from the 'centre' of the cluster slightly to the left, in line with denser regions of the skewed populations (Figures 5.7E, 5.8E and 5.9E).

5.3.2.1 Total events 10^4

In the dataset with 10^4 total events (Figure 5.7), highly similar patterns of performance were seen compared with the rare-normal dataset; none of the software were able to detect the 10-cell cluster (0.1% rare frequency), then at the 50-cell level (0.5%), cluster detection was achieved by Flock2 (52.3 ± 1.5), FlowSOM (51.3 ± 1.5) and SWIFT (48.3 ± 1.2). Only at the 500-cell level (5%) was SPADE3 able to detect the rare cluster with good accuracy and repeatability (511.7 ± 18.5), and PhenoGraph was unable to detect any of the rare

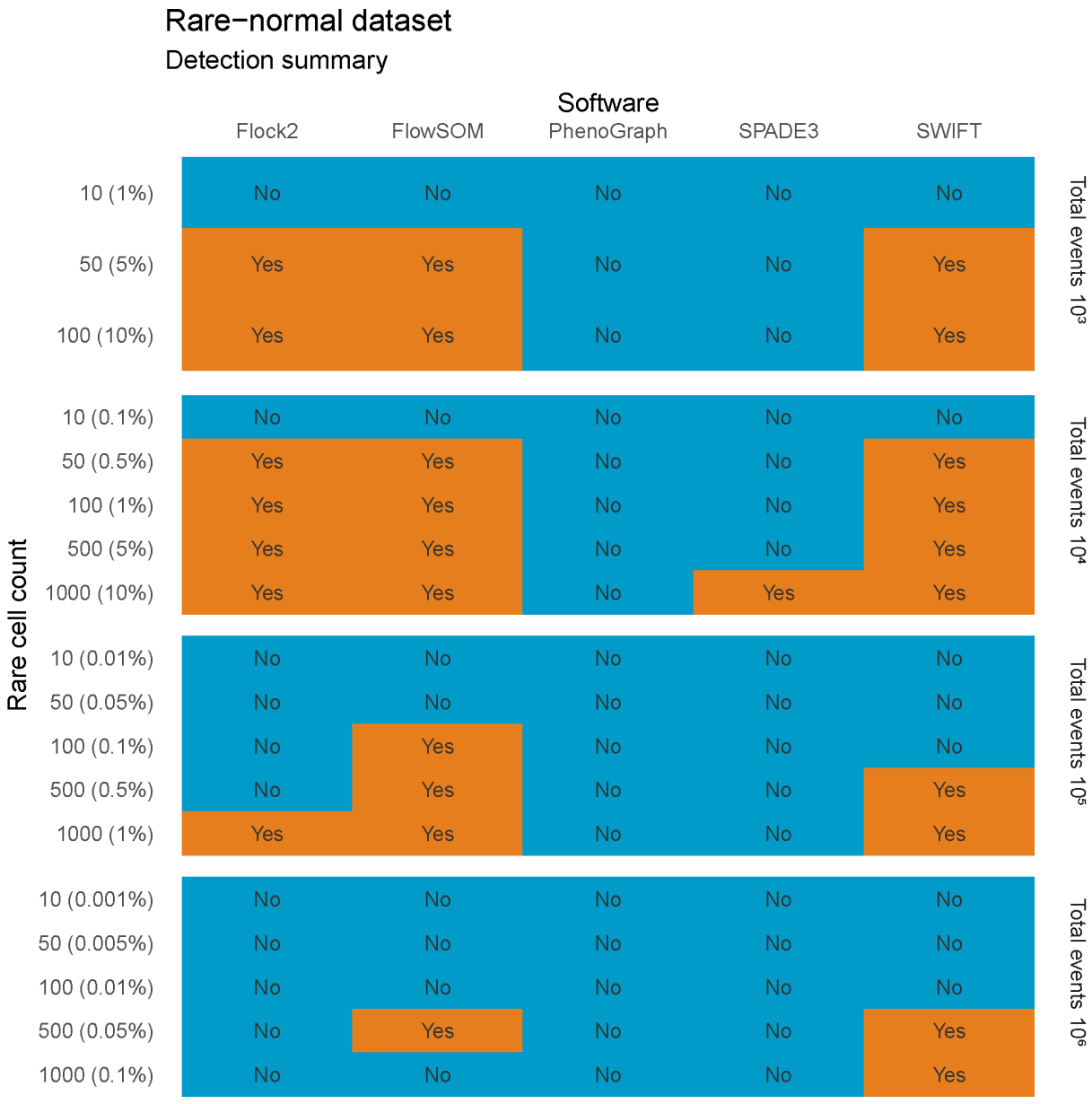


Figure 5.6: Heatmap summarising software performance in rare cell detection (normally distributed clusters), based on a detection count threshold of $\pm 20\%$.

clusters. The presence of rare-skew populations in this dataset did not appear to affect the software performance.

5.3.2.2 Total events 10^5

None of the software were able to identify the 10-cell rare population among 10^5 events (0.01%) (Figure 5.8). At the 50-cells level (0.05%), as previously seen in the rare-normal dataset, FlowSOM was once again the slightly better performing software amongst others at rare cell detection (76.3 ± 13.2). Flock2 and FlowSOM managed to identify the 100-cell cluster (0.1% rare frequency) with good accuracy and repeatability, with counts of 105.3 ± 12.1 and 103.3 ± 3.2, respectively. Both PhenoGraph and SPADE3 failed to detect the rare-skew clusters in this dataset.

5.3.2.3 Total events 10^6

At 10^6 total events (Figure 5.9), the rare-skew clusters of 10 (0.001%) and 50 events (0.005%) were not detected by any software, however FlowSOM was able to identify the 100-event cluster (0.01%) with a count of 117.7 ± 12.3. The 500-event cluster (0.05%) was detected by Flock2 (527.7 ± 27) and FlowSOM (501.7 ± 0.6). Once again, none of the clusters were identified by PhenoGraph and SPADE. Interestingly, SWIFT was unable to detect any of the rare-skew clusters at this level, but in the rare-normal dataset it had been able to detect the 500-cell cluster, suggesting that the presence of skewed populations negatively affected its performance.

5.3.2.4 Performance analysis

Results from the rare-skew runs were similar to the rare-normal ones. Again, no software was able to detect the 10-cell cluster. The LoD typically improved in rare cell percentage terms as the magnitude of total events increased (Figure 5.10). For example, with FlowSOM, the LoD improved from 0.5% to 0.05% and then to 0.01% as the total events increased from 10^4 to 10^5 and 10^6 in the rare-skew dataset. The introduction of non-normal distributions mainly affected SWIFT, with detection performance deteriorating notably at the 10^6 level compared to runs with normally-distributed clusters in Figure 5.3. Conversely, performances of Flock2 and FlowSOM appeared to improve at the same level with skewed clusters, possibly because of a sparser density of points between the two clusters in the dataset. Using the example detection threshold of $\pm 20\%$ (Figure 5.11), the results from the synthetic two-cluster rare cell datasets suggested that FlowSOM was the best performing computation tool for rare-skew cell detection out of the

Figure 5.7: Clustering examples from different software on a synthetic two-cluster rare cells dataset, containing 10^4 total events, with skewed clusters.

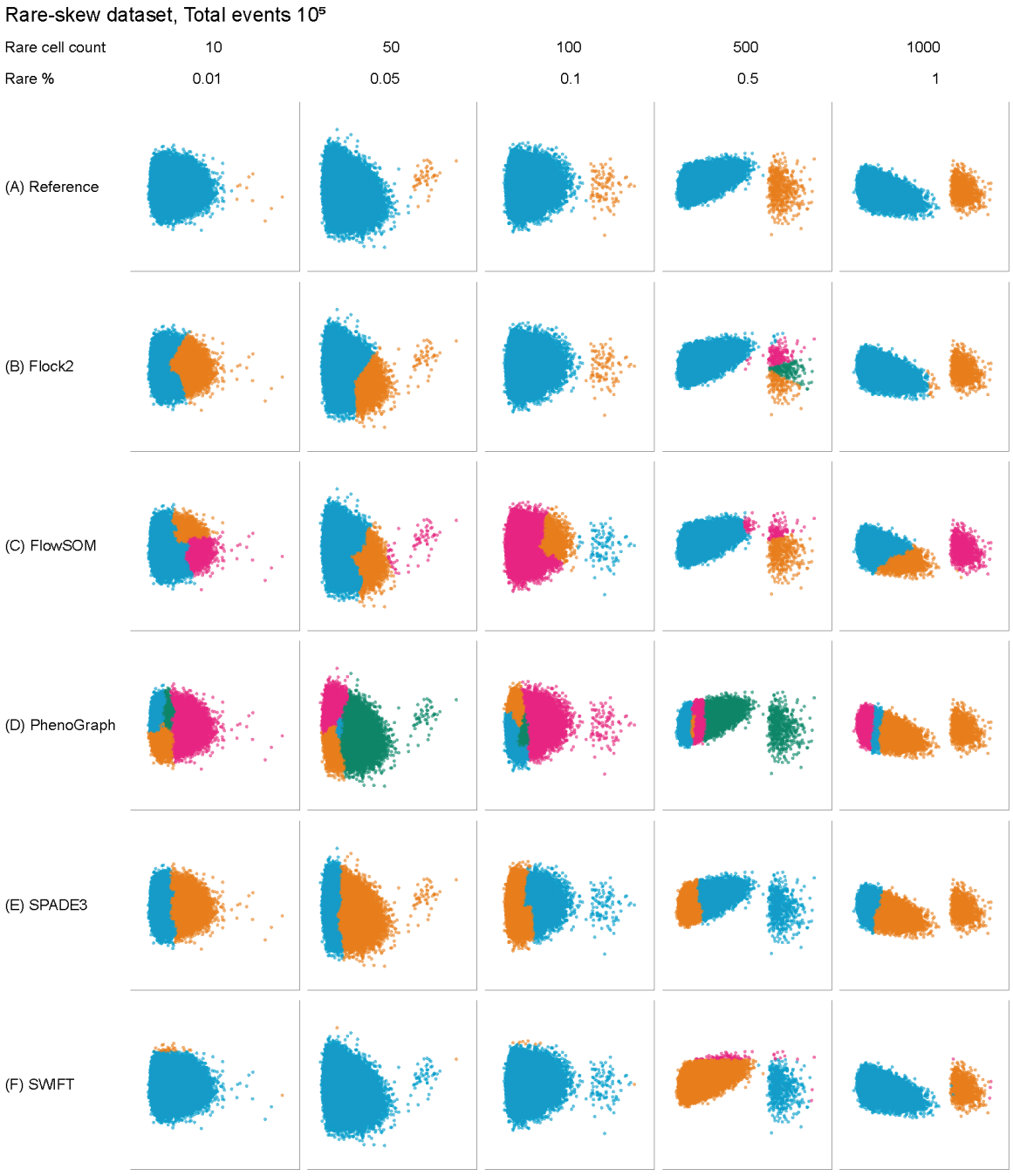


Figure 5.8: Clustering examples from different software on a synthetic two-cluster rare cells dataset, containing 10^5 total events, with skewed clusters.

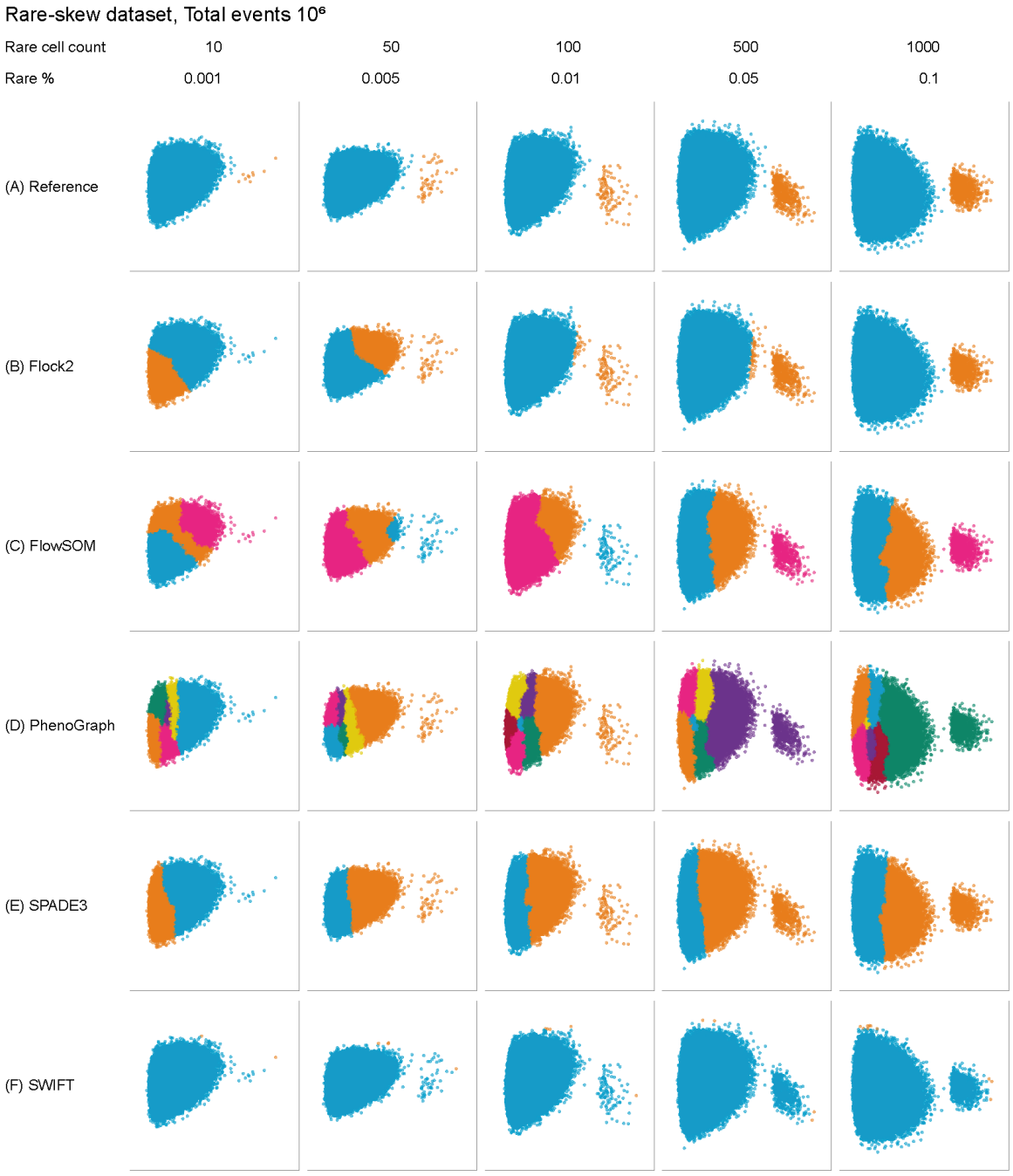


Figure 5.9: Clustering examples from different software on a synthetic two-cluster rare cells dataset, containing 10^6 total events, with skewed clusters.

five tested here. This was followed by Flock2, SWIFT, SPADE3, and lastly, PhenoGraph.

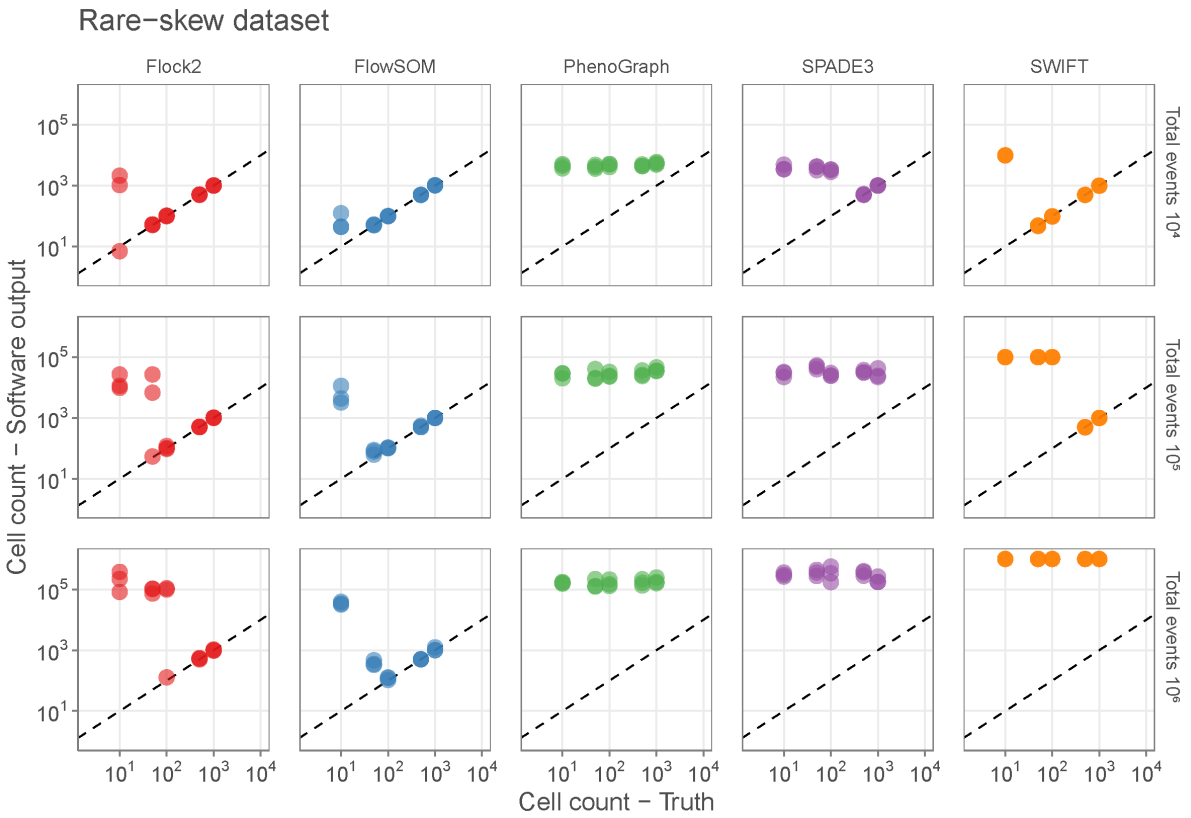


Figure 5.10: Plots of rare-skew cell count truth vs. software output across different software (columns) and increasing levels of total events (rows). Points on the dashed line indicate a 'hit'.

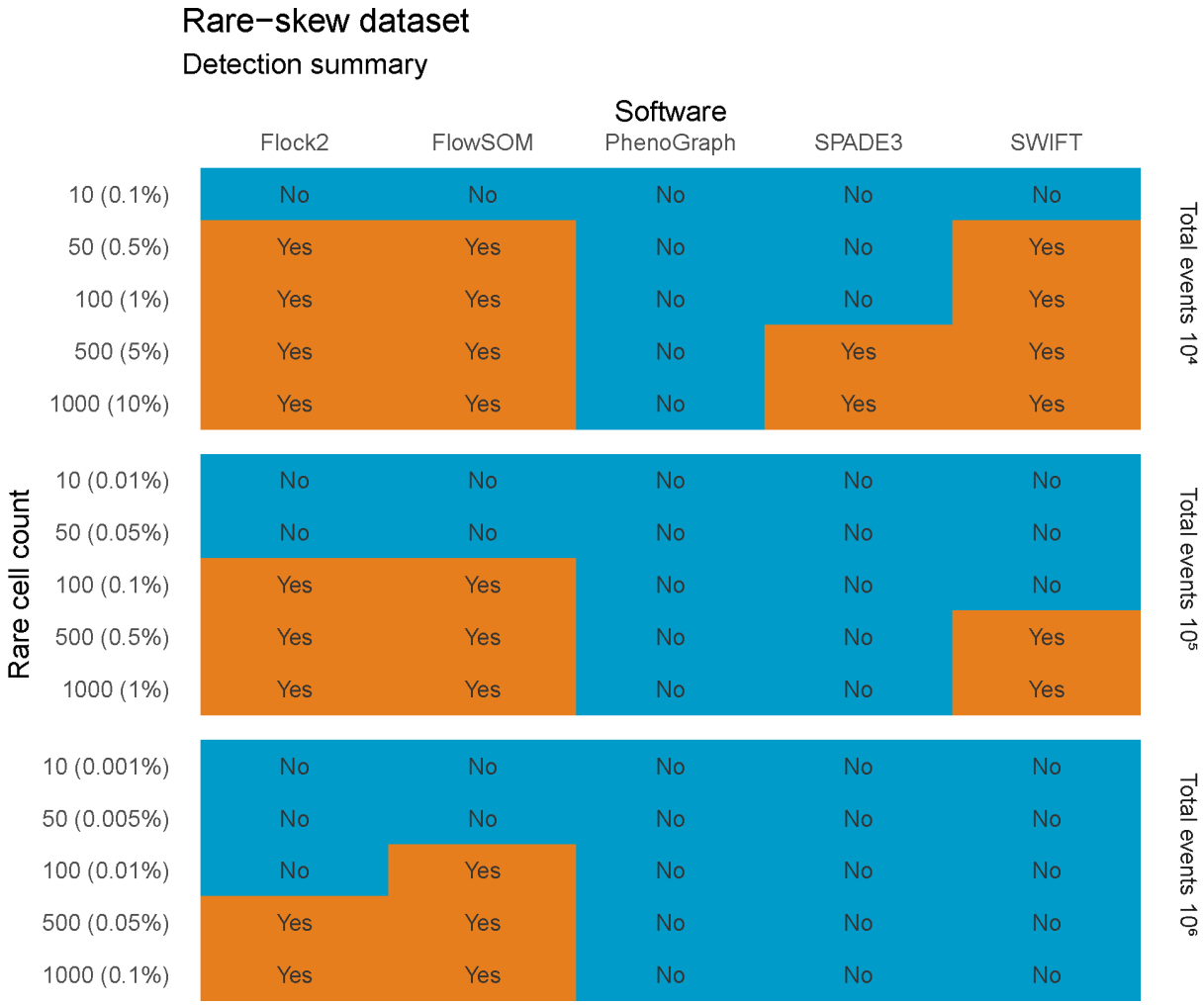


Figure 5.11: Heatmap summarising software performance in rare cell detection (skewed clusters).

5.3.3 Results of CD34 dataset runs

Real cell datasets were next used to assess the software ability to detect rare cells from experimental flow cytometry data, and also to validate the results seen in the runs with synthetic datasets. Firstly, software were challenged to identify a rare CD34+ population from a dataset of mobilised peripheral blood. This dataset followed the course of a CD34+ stem cell population through a 21-day culture period, which was abundant at day 0 but gradually became rarer over time, as the cells matured and their pluripotency diminished (Figure 5.12). Specifically, data from days 15, 18 and 21 were selected for rare cell analysis.

Comparison of the manual gate reference against the rare cell detection outputs from software demonstrated most software were able to detect the target population at day 15, with the exception of PhenoGraph (Figure 5.13). By day 18 and day 21, most software struggled to identify the target population, with only limited runs from SPADE1 and SWIFT being successful.

5.3.3.1 Day 15

The number of rare events (manually gated) at day 15 was 41.0 ± 9.6 (1.5%). SPADE3 and FlowSOM successfully detected the rare population with reasonable accuracy and repeatability, returning counts of 49.0 ± 8.7 and 48.7 ± 20.2, respectively (Figure 5.14, top row). SPADE1 reported slightly lower counts of 29.3 ± 6.7. SWIFT detected the rare population in all three runs, however one output included a number of false positive events that reduced its performance to counts of 59.0 ± 33.1. Flock2 detected the rare population in 2 out of 3 replicates, reporting mean counts of $831 \pm 1,368$ noting the SD is a very big number. PhenoGraph was the only software that failed to detect the rare cluster for all 3 replicates in this condition.

5.3.3.2 Day 18

The number of rare events (manually gated) decreased to 6.3 ± 2.1 (0.2%) (Figure 5.14, middle row). SWIFT detected the rare cluster with reasonable accuracy and repeatability, giving counts of 9.3 ± 4.9. SPADE1 performed poorly, with counts of 77.7 ± 108.8. Flock2, FlowSOM, PhenoGraph, and SPADE3 did not accurately detect the rare events in this condition.

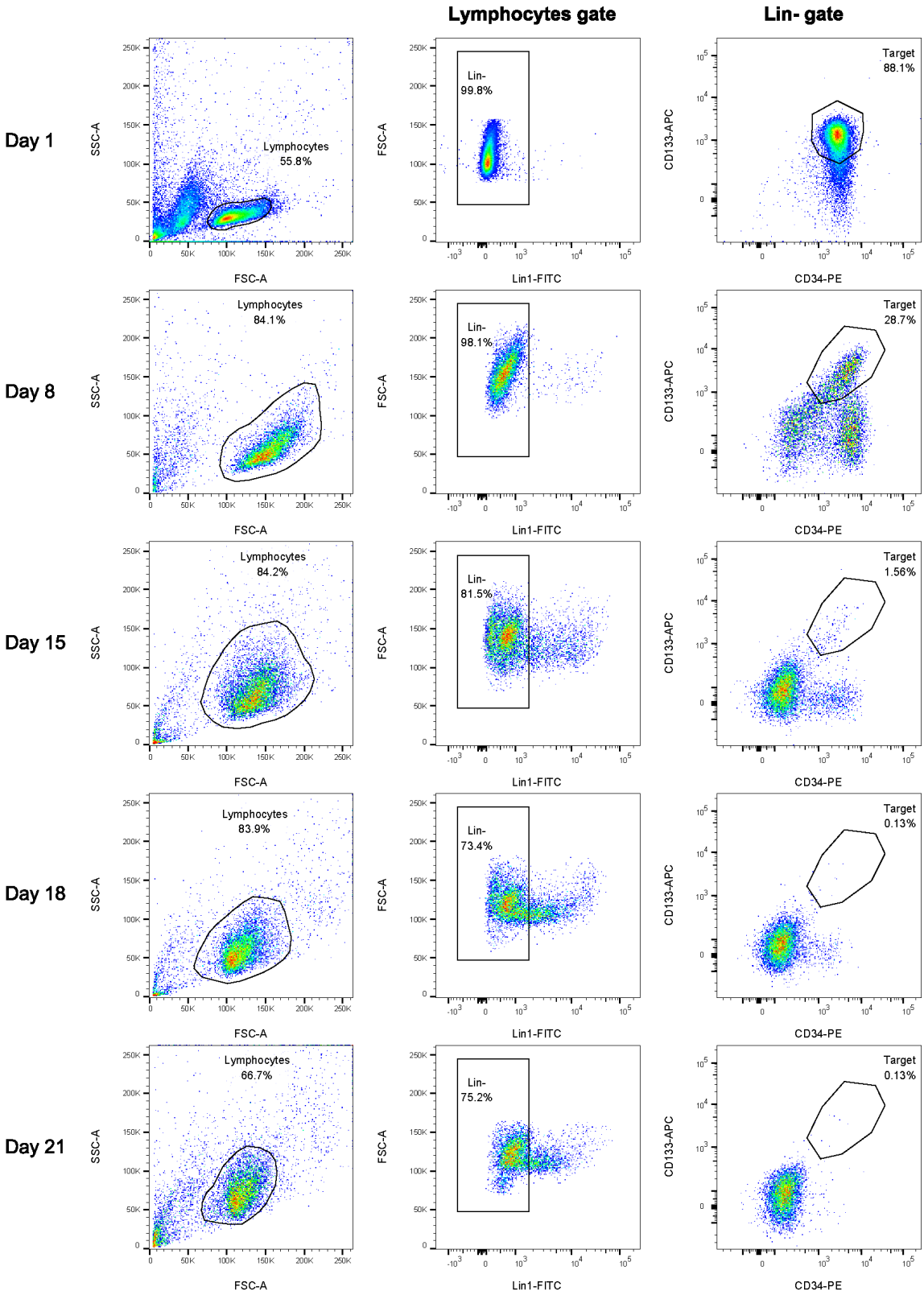


Figure 5.12: Representative flow cytometry plots and manual gating strategy of the CD34 dataset, showing the target population (Lin-/CD34+CD133+) decreasing in number and becoming rarer over time from day 1 to day 21.

Figure 5.13: Comparison of software detection of CD34+ rare cell population.

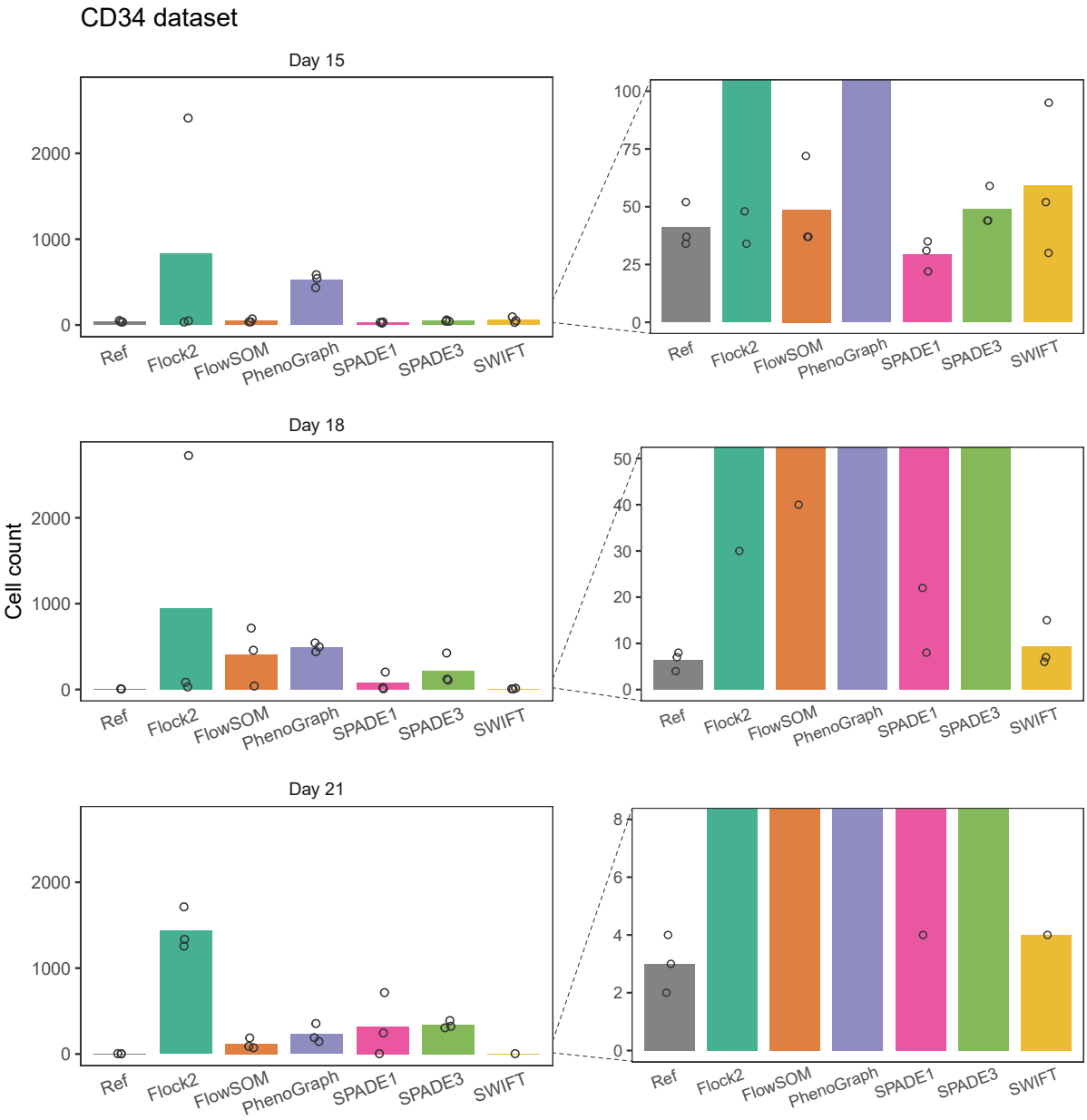


Figure 5.14: Comparison of rare cell counts of software outputs to reference values (grey bars) for the CD34 dataset at days 15, 18 and 21. Bars, mean; circles, individual outputs.

5.3.3.3 Day 21

The day 21 dataset proved to be the most challenging condition for the software, because of the low number of rare (manually gated) events at 3.0 ± 1.0 (0.1%) (Figure 5.14, bottom row). Two of the software tested were partially successful; SPADE1 detected the rare events in 1 out of 3 replicates, thereby giving poor repeatability results with counts of 321 ± 362, and, SWIFT also detected the rare events in 1 out of 3 replicates, but did not give any cluster partitions for the other two resulting in a lack of outputs for further statistical analysis. All the rest of the software tested (Flock2, FlowSOM, PhenoGraph, and SPADE3) failed to detect the rare events in this dataset.

5.3.3.4 Note on pooled runs

It was noted during software runs that SPADE1 and SPADE3 gave user options to pool several files from an experiment together before running the clustering algorithm. This differed from most other analysis tools, where each file is run individually without pooling. This pooled function was tested by combining all files at multiple timepoints from day 0 to day 21 together. The accuracy and repeatability results of rare cell detection using this pooled strategy outperformed those where individual files were run (Figure 5.15). In this instance, the reason for the improvement was potentially because the abundance of the target CD34 population at earlier timepoints acted as a guide to their rare detection at later stages. This pooling method does not directly equate to high performance of rare cell detection, nevertheless, users with relevant datasets may find it useful.

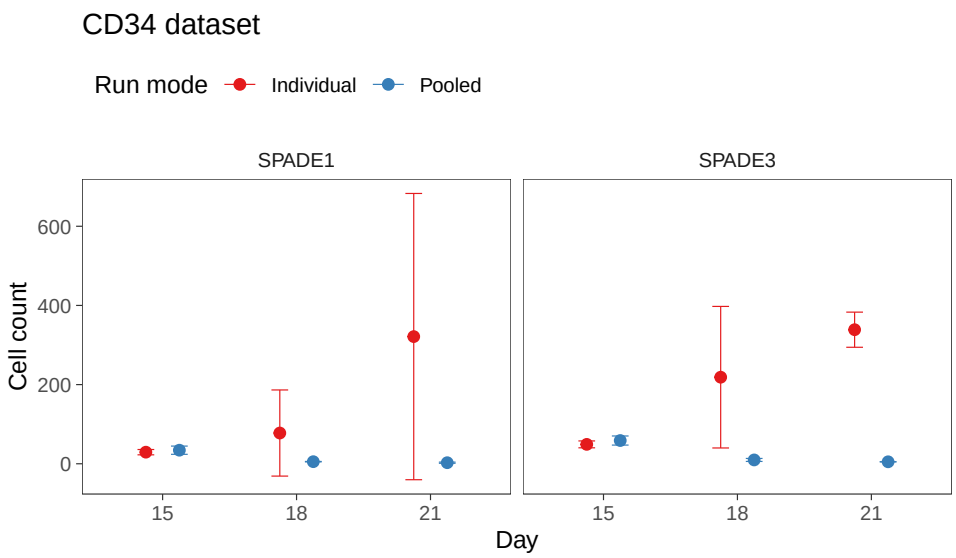


Figure 5.15: Comparison of cell count outputs from CD34 dataset runs using individual against pooled mode in SPADE1 and SPADE3. N=3, mean ±1SD.

5.3.4 Results of K562 dataset runs

While the CD34 dataset provided rare populations of low absolute counts, the magnitude of total events (10^3) did not cover the same range as the synthetic datasets generated. In this regard, the K562 dataset was applied for comparable analysis because the total number of events after data cleaning and pre-processing amounted to approximately 9×10^5 cells. This dataset consisted of a main population of K562 cell, spiked-in with limiting dilutions of K562 GFP subpopulations to simulate rare cells (as described in Section 5.2.2.2).

5.3.4.1 General observations

The K562 dataset contained well-separated GFP- and GFP+ populations, with slight skew distributions 5.16. Comparison of the manual gate reference against the software outputs showed that Flock2 and SWIFT were able to partition the dataset along the GFP channel into the two expected clusters (Figure 5.17). The rare cluster was also successfully isolated by FlowSOM, which showed patterns of overclustering in the main GFP- population. A striking observation was the horizontal partitioning of the dataset by PhenoGraph, SPADE1, and SPADE3, where the main population of GFP- cells along with the rare GFP+ cells were spliced along the side scatter parameter rather than the GFP channel. These three software were unable to separate the rare population from the data.

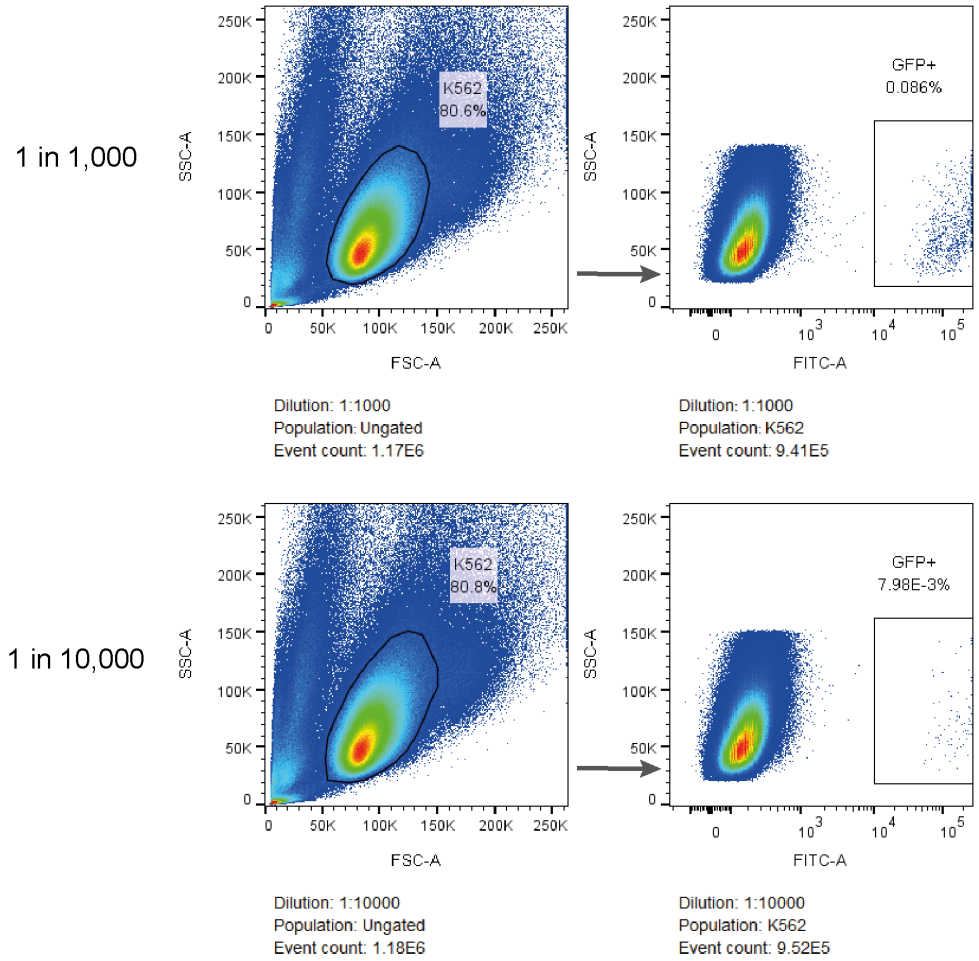


Figure 5.16: Representative flow cytometry plots and manual gating strategy of the K562 dataset, showing the 'spiked-in' rare GFP+ cell population at 1 in 1,000 and 1 in 10,000 dilution conditions.

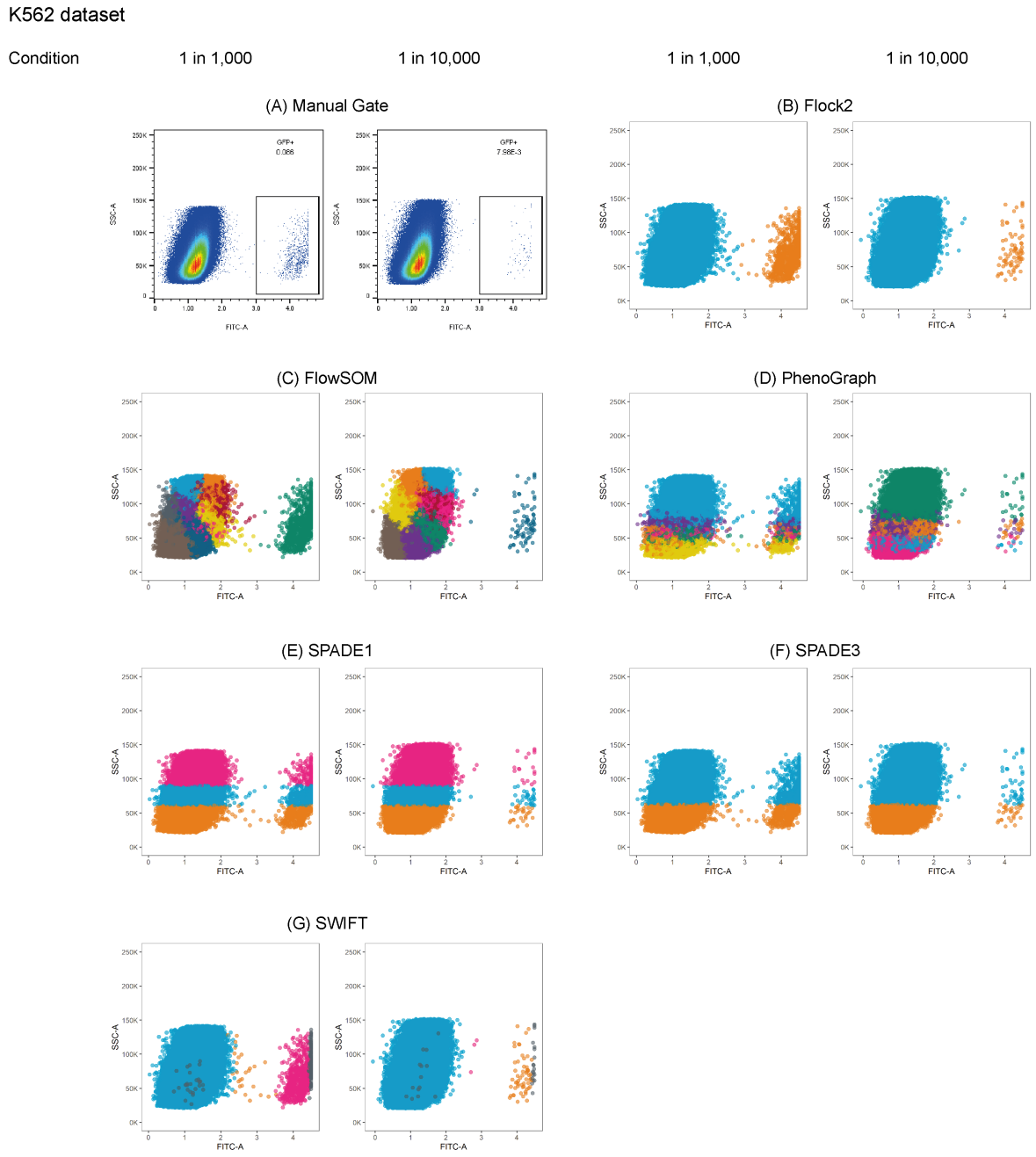


Figure 5.17: Comparison of software detection of GFP+ rare cell population in the K562 dataset.

5.3.4.2 1 in 1,000 dilution

In the 1 in 1,000 dilution level dataset, the number of target events manually gated was 774 ± 129, giving rare frequencies of approximately 0.086% (Figure 5.18, top row). FlowSOM was able to perfectly match the manually gated data with rare cluster counts of 774 ± 129. This high performance was closely followed by Flock2, which was able to detect the rare population as counts of 746 ± 131. SWIFT slightly underestimated the rare cluster, returning counts of 638 ± 104. This under-reporting was most likely due to a function of SWIFT's algorithm that discarded events on the axis limits that do not fit a Gaussian distribution. PhenoGraph, SPADE1 and SPADE3 performed poorly, being unable to detect the rare cluster with counts of over one order of magnitude above the target. The SPADE algorithm appeared to have ignored the rare cluster as noise.

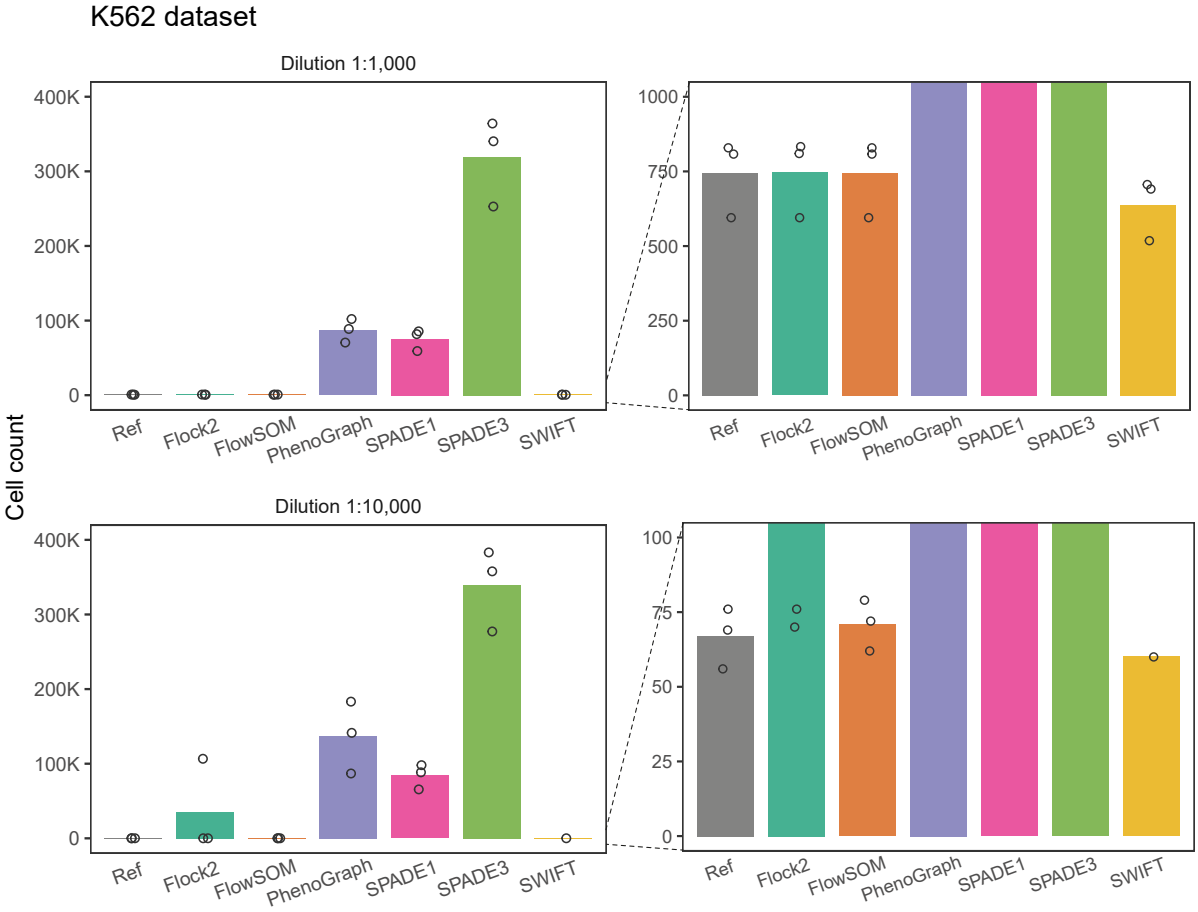


Figure 5.18: Comparison of rare cell counts of software outputs to reference values (grey bars) for the K562 dataset at the 1 in 1,000 and the 1 in 10,000 conditions. Bars, mean; circles, individual outputs.

5.3.4.3 1 in 10,000 dilution

In the 1 in 10,000 dilution level dataset, the number of target events manually gated was 67 ± 10, giving rare frequencies of approximately 0.008% (Figure 5.18, bottom row). FlowSOM was able to detect the rare cluster from three replicates with a high level of accuracy and repeatability, reporting cell counts of 71 ± 8.5. Flock2 accurately detected the rare population for two of the replicate datasets, however, it failed to detect the rare cluster for the third file, returning over 1×10^5 false positive cells as a result of partitioning the data along the side scatter channel instead of the GFP channel. This shortcoming in repeatability resulted in a poor mean cell count of over 35,000. Repeatability issues also appeared in SWIFT, which managed to detect the rare cluster for only one of the replicates (60 cells). For the other two replicates, SWIFT was completely unable to identify the rare cluster and did not return the required number of clusters for further analysis and calculation of performance metrics. Similar to the 1 in 1,000 condition, PhenoGraph, SPADE1, and SPADE3 were unable to detect the rare cluster and reported inaccurate cell counts over three orders of magnitude away from the manual reference values. All three software displayed the characteristic partitioning along the side scatter channel instead of the GFP channel, indicative of poor performance.

Taken together, the software runs from the K562 titration dataset showed FlowSOM to be the best performing rare cell detection tool out of the six tested here in terms of accuracy and repeatability. Flock2 and SWIFT also performed reasonably well, however both their repeatability began to fail at the higher rarity level. PhenoGraph, SPADE1 and SPADE3 failed to detect any of the rare clusters from the K562 datasets.

5.3.5 Comparison between synthetic and real cell datasets

From both synthetic and real cell runs, common themes that emerged were that absolute rare counts below 10 cells were not reliably detected by most software. For the larger dataset with 10^4 to 10^6 total events, FlowSOM, Flock2 and SWIFT were better able to detect rare populations. However, repeatability deteriorated for Flock2 and SWIFT as the rarity increased. SPADE3 was able to detect a rare cluster with a frequency of 5% in the synthetic datasets, but not at 1%. In comparison, a rare population of 1.5% in the CD34 day 15 dataset was detected by both SPADE1 and SPADE3, but any lower frequencies were not detected. Therefore, a rare frequency of 1% may potentially be the detection limit threshold for SPADE algorithms run on default settings.

Analysis of the lower limits of detection (LLoD) between synthetic (normally dis-

tributed) and real world rare cell datasets showed FlowSOM and SWIFT were able to detect the rare populations at or below 5% and 0.1% for total events of 10^3 and 10^6, respectively (Table 5.4). Flock2 detection performance was not consistent across datasets; while the synthetic rare population was detected with a LLoD of 5% at the 10^3 level, the analogous population was not detected for the real cell CD34 dataset. Similarly, at the 10^6 level, Flock2 was not able to detect any of the synthetic rare cell populations, but managed to a real cell cluster in the K562 dataset with a LLoD of 0.086%. Inconsistencies were also observed for SPADE3 (at 10^3 total events), and comparisons could not be drawn from SPADE1 because the platform failed to process synthetic datasets. Finally, PhenoGraph was unable to detect the rare population across all synthetic and real cell datasets tested. Overall, the level of concordance between these LLoD results suggest application of synthetic datasets is a viable method of testing automated flow cytometry data analysis software.

5.3.6 Comparison of metrics

This study on rare cell detection was extended to explore the different evaluation metrics available to describe software performance. This particular exercise focussed on the larger datasets (10^6) in both synthetic and real-world cases (rare-skew and K562, respectively), and the rare cell conditions within each dataset already identified at the critical region near the LoDs of software. Evaluation metrics considered were those from binary classification: accuracy, precision, sensitivity (also known as recall), specificity, and the F1 score (defined in Chapter 3); as well as the CV.

The comparison of the metrics for each software at selected rare cell conditions are shown in Figure 5.19 for the Rare-skew dataset, and Figure 5.20 for the K562 dataset. These results can be analysed alongside those from Figure 5.10 and Figure 5.18. For the binary classification metrics, a higher score indicates better performance, whereas for the CV, lower is better. The results give a mixed picture in the agreement of the metrics to each other.

The accuracy metric measures the proportion of correctly identified rare and non-rare events out of the total population. The utility of this metric is limited given the uneven sizes of clusters in these datasets, however an overview of the software performance can be demonstrated. FlowSOM scored highly on accuracy when analysing the rare-skew dataset, while SWIFT gave low accuracy (Figure 5.19A). For the K562 dataset, accuracy scores for Flock2, FlowSOM and SWIFT were > 0.95, and SPADE3 gave notably lower accuracy score compared with the other software (Figure 5.20A).

Table 5.4: Lower limits of detection of software for runs on synthetic and real world flow cytometry datasets. ND, Not detected.

Total events	Software	Lower limit of detection			
		Rare-normal dataset	Rare-skew dataset	CD34 dataset	K562 dataset
10^3	Flock2	5%	-	ND	-
	FlowSOM	5%	-	1.5%	-
	PhenoGraph	ND	-	ND	-
	SPADE1	-	-	1.5%	-
	SPADE3	ND	-	1.5%	-
	SWIFT	5%	-	0.2%	-
10^4	Flock2	0.5%	0.5%	-	-
	FlowSOM	0.5%	0.5%	-	-
	PhenoGraph	ND	ND	-	-
	SPADE1	-	-	-	-
	SPADE3	10%	5%	-	-
	SWIFT	0.5%	0.5%	-	-
10^5	Flock2	1%	0.1%	-	-
	FlowSOM	0.1%	0.1%	-	-
	PhenoGraph	ND	ND	-	-
	SPADE1	-	-	-	-
	SPADE3	ND	ND	-	-
	SWIFT	0.5%	0.5%	-	-
10^6	Flock2	ND	0.05%	-	0.086%
	FlowSOM	0.05%	0.1%	-	0.008%
	PhenoGraph	ND	ND	-	ND
	SPADE1	-	-	-	ND
	SPADE3	ND	ND	-	ND
	SWIFT	0.05%	ND	-	0.086%

The precision metric shows the proportion of correctly predicted rare cells out of the total predicted rare events, including any false positives (FPs). With the exception of FlowSOM at the 100-rare cell level, the majority of rare-skew software runs incorrectly assigned numerous non-target events to the rare cluster, resulting in low precision scores (Figure 5.19B). The high number of FPs identified in the target cluster by PhenoGraph, SPADE1 and SPADE3 reduced the precision score for the K562 dataset runs, in contrast, Flock2 and FlowSOM predicted very few FPs, and scored highly (Figure 5.20B).

The sensitivity metric measures the proportion of true positives that are correctly identified by the software, and was scored highly for all software on the rare-skew dataset (Figure 5.19C), most likely because of manual interpretation of clusters that selected for ones incorporating the rare events — an element of operator bias/ subjectivity that cannot be removed from automated data analysis. Software runs from PhenoGraph, SPADE1 and SPADE3 that partitioned the K562 data horizontally (thereby complicating manual interpretations) reported low sensitivity scores (Figure 5.20C).

Specificity is perhaps not a useful metric for evaluating rare cell detection because it focuses on the abundant non-target population. Its values are similar to accuracy scores because both calculations include the large condition negative value in the denominator (Figure 5.19D, Figure 5.20D).

The F1 score is the harmonic mean of precision and sensitivity (recall), and was the metric of choice for previous comparison studies [49, 47], and therefore has been included here for completeness (Figure 5.19E, Figure 5.20E).

The CV measures the repeatability of the software runs, although it can be misleading when reported on its own without context on accuracy (trueness), as shown for SWIFT, which gave low CV scores among consistently inaccurate outputs from the rare-skew dataset (Figure 5.19F). Most software in the K562 runs reported CVs below 40%, except for Flock2 in the 1:10,000 condition (CV = 173%), and SWIFT, which only completed 1 in 3 runs so the CV could not be calculated (Figure 5.20F).

Taken together, this comparison of performance metrics has shown there is no single value that can be used to entirely evaluate the performance of software, and has highlighted the difficulties in selecting suitable metrics for reporting purposes. For rare cell detection, the utility of one metric may have more weight than another, and certain metrics can be misleading when taken out of context. Derivative metrics that summarise performance into a single value potentially lose valuable information originating from differing sources. Furthermore, basic statistics on the cell count mean and SD may be more intuitive for users to understand the quality of software analysis. The full interpretation of these metrics would require a given threshold for determining 'acceptable' or 'unaccept-

Table 5.5: Software rankings for the rare-skew 10^6 dataset.

Software	Accuracy	Precision	Sensitivity	Specificity	F1	CV
Flock2	2	2	4	2	2	5
FlowSOM	1	1	3	1	1	2
PhenoGraph	3	3	1	3	3	3
SPADE3	4	4	1	4	4	4
SWIFT	5	5	5	5	5	1

Table 5.6: Software rankings for the K562 dataset.

Software	Accuracy	Precision	Sensitivity	Specificity	F1	CV
Flock2	3	3	2	3	3	4
FlowSOM	1	1	1	1	1	6
PhenoGraph	5	6	6	5	6	3
SPADE1	4	4	5	4	4	1
SPADE3	6	5	4	6	5	2
SWIFT	2	2	3	2	2	5

able' software performance, which would vary depending on the scenarios the software tools are applied in.

The rankings for each software according to the different evaluation metrics are summarised in Table 5.5 for the rare-skew 10^6 dataset, and Table 5.6 for the K562 dataset. The tables show software that rank highly in one metric can rank lower in another. No software maintained the same position across all performance metrics. The rankings reveal that FlowSOM outperformed other software in both datasets on all metrics except sensitivity and CV.

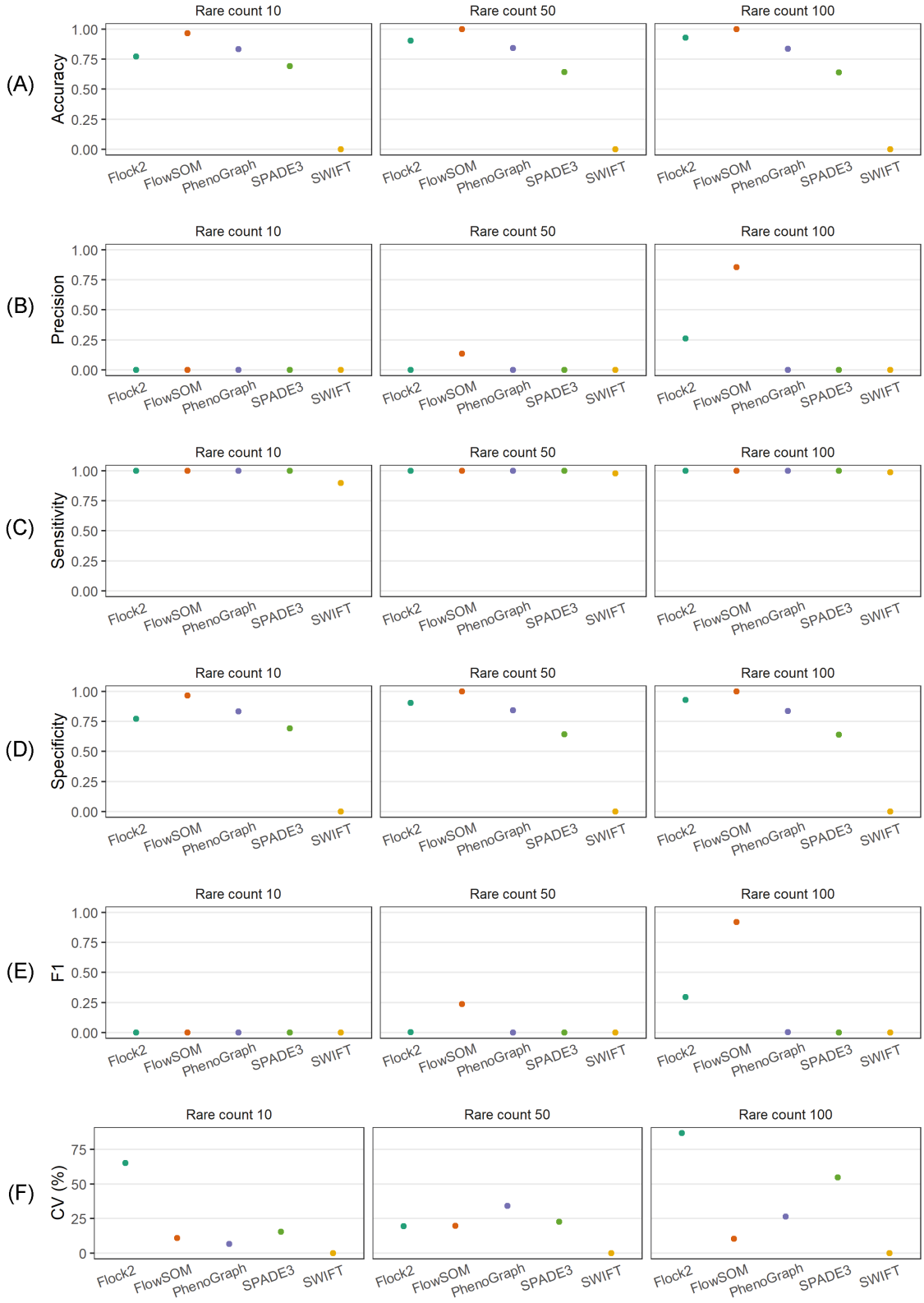


Figure 5.19: Evaluation metrics for software runs on the rare-skew 10^6 dataset at selected rare cell levels.

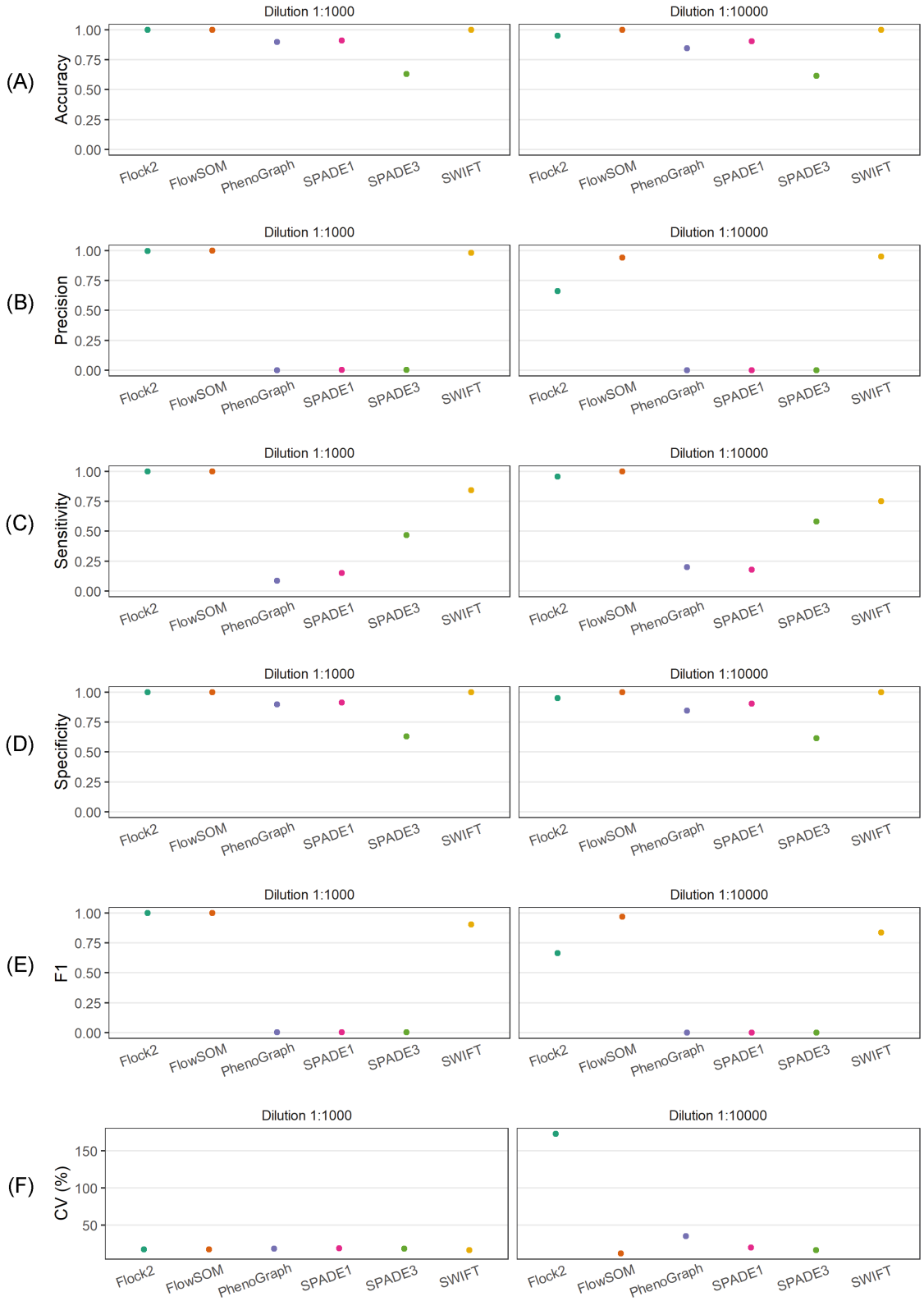


Figure 5.20: Evaluation metrics for software runs on the K265 dataset.

5.4 Discussion

Rare cell detection is an important task in flow cytometry, and a number of automated data analysis tools are available that have the capacity to eliminate operator variability in identification of rare events. Currently, a lack of standardisation and user guidance exists on the use of automated software to identify rare cell populations. There is scope in the field for comparison studies of software, using well-designed synthetic datasets.

In this chapter, synthetic datasets have been applied to evaluate the performance of several automated flow cytometry data analysis software, specifically in the task of rare cell population detection. The performance evaluation results from synthetic datasets have then been validated using real cell dataset examples.

The results from this chapter recapitulate findings from previous chapters, which indicated considerable variation in outputs between different software analysing the same synthetic dataset. The variability is typified in the rare-normal dataset with 10^4 total events, where the LoD ranged from 50 cells (0.5%) in Flock2, FlowSOM and SWIFT, to 500 cells (5%) in SPADE3, and undetected from PhenoGraph. Limits of detection were found to vary as a function of total events in the dataset, generally improving in percentage terms as the magnitude of total events increase. Absolute rare counts below 10 cells were not reliably detected by most software. Repeatability deteriorated as the rarity increased for Flock2 and SWIFT.

For users, the results illustrated in this chapter provide a clearer understanding of the detection limits of different software and their suitability for rare cell analysis. Thereby enabling decisions to implement these automated tools in analysis pipeline based on their own data.

Results here confirm previous work showing SWIFT outperformed other tools at identifying less frequent populations ($< 0.1\%$) [3]. But importantly, this study highlighted how SWIFT performance at this level was not maintained in the presence of skewed populations.

The comparison of several performance metrics in this study demonstrated the lack of agreement between metrics for describing concepts of accuracy in software performance, and therefore careful consideration needs to be given to derive a suitable range of metrics for judging the strengths and limitations of automated analysis software.

The rare populations from synthetic datasets have a known ground truth based on mathematical principles that give advantages over manually gated references, which are based on operator judgement, potentially biased and not reproducible. The sizes of rare populations can be specifically controlled, as well as the total sample size. This is not

straightforward with real cells and flow cytometry experiments. Acquisition of rare events from large, multidimensional datasets ($\sim 10^6$) require extensive cell starting materials, reagents, time and resources. Using cell sorters to dispense exact numbers of cells into a sample is a possibility, however that approach introduces its own sources of variation.

Given the possibility that rare clusters were disregarded as noise by these algorithms, the fine tuning of user parameters may improve performance. However, the iterative nature of optimising inputs from each individual algorithm demands extensive time and resources, so was outside the scope of this study but may be an option for further work.

It was beyond the scope of this study to addressed inclusion of negative gating controls (e.g. unstimulated, FMO, and isotype controls), which is how rare populations are routinely gated in the lab following best practice guidelines [205]. This is mainly because automated approaches that apply thresholds based on external data to gate rare cells do not fall into the same category of clustering algorithms investigated here (see [206] for automated approach to set objective thresholds for rare event detection).

The work presented here suggest the current field of flow cytometry automated clustering tools have not yet reached maturity in rare population detection. Combining the clustering tools together with other algorithms, such as dimensionality reduction, together in an analysis pipeline, or even in combination with supervised learning methods that require large banks of training data, may be potential solutions. With the growing variety of possible combinations of analytical methods, it will become increasingly important to develop standards to harmonise working practices across the healthcare and biomanufacturing industries.

5.5 Chapter conclusions

- Large two-cluster synthetic datasets ($10^4 - 10^6$) have been generated with rare cell populations down to 0.001% frequency, with both normal and skewed distributions.
- The synthetic datasets generated have been applied to evaluate the rare cell detection performance of various software that utilise different clustering algorithms: Flock2, FlowSOM, PhenoGraph, SPADE1, SPADE3, and SWIFT.
- FlowSOM and SWIFT were the best performing methods for the normally distributed dataset, followed by Flock2. This outcome was validated by runs from the K562 real cell dataset.
- Analysis using the rare-skew dataset revealed a deterioration in SWIFT performance compared with rare-normal clusters.
- The experiments performed here highlighted the lack of readiness of a number of

software for commercial use; SPADE1 and SPADE3 performed poorly for low frequency populations ($< 1\%$), and in particular, PhenoGraph failed to detect any of the rare populations from all datasets.

- Results from the synthetic datasets suggested that an absolute rare event size of 10 cells (irrespective of total event size) was beyond the limits of detection for most software. This finding was supported by the runs from the CD34 real cell dataset.
- A range of performance metrics were explored and found to have varying utilities for describing the confidence in rare cell detection by automated software.
- Overall, this work has demonstrated the robustness of synthetic datasets in performing critical assessment of automated flow cytometry data analysis software, specifically in the detection of rare cell populations.

Chapter 6

Noise

6.1 Introduction

The development of synthetic flow cytometry datasets with controlled separation between clusters and skewed population distributions (Chapter 4) and rare cell populations (Chapter 5), followed by their application to evaluate different software for automated population identification, have demonstrated the variability in outputs of different clustering algorithms, along with their strengths and limitations.

The utility of synthetic datasets relies on their fair representation of real-world data, thus, if they are to be applied as reference materials, users can be confident of their fitness for purpose. In practice, not all features of flow cytometry data can be comprehensively covered by their synthetic counterparts.

It has been necessary to model 'clean' datasets to date, however it is recognised that real-world flow cytometry data is typically 'noisy'. With this in mind, this Chapter extends the synthetic datasets generated so far by introducing noise properties, so that the response of software tools to noise can be better understood.

A further rationale to optimise and expand synthetic characteristics to include noise is that the specific, controlled generation of noise in real data is inherently difficult to achieve, and may be viewed as an impractical use of precious biological samples.

Previous work to generate artificial noise have simulated cell debris by addition of random events at variable proportions from 1% to 50% [207, 208]. Noise layers of uniform distributions and normal distributions have been applied [82, 77]. In the field of mass cytometry, a data pre-processing step offered by device software termed 'randomisation' manipulates the distribution of acquired data, that in effect injects random noise into it to improve visualisations. This process has been shown to negatively affect high dimensional

data analysis leading to misinterpretations of data, and is not recommended [209]. Along these lines, to solve the appearance of "striped" flow cytometry data after compensation [210, 170], very small amounts of noise (standard deviation 0.003) have been added to data for better visualisations [211].

In this study, noise is artificially introduced into flow cytometry synthetic datasets in a systematic way, in order to investigate the robustness of computational tools to noise components. Compared with previous work, this study benefits from the absence of potential bias from developers of software.

This chapter begins by examining the definitions and sources of noise in the context of flow cytometry. The design elements of a dataset with noise are next considered, and generated. Then, the noise datasets are processed through several automated algorithms, and the effect of noise on algorithm performances are evaluated. This chapter concludes with a summary and directions for further research.

6.1.1 Chapter aims

The aims of this chapter are to:

- Inject noise elements into the synthetic datasets generated so far to create noise-separation, noise-skew and noise-rare datasets.
- Process the noisy datasets through different automated cell population identification software.
- Evaluate the accuracy and repeatability of software outputs to understand the effect of noise on software performance.

6.2 Sources of noise in flow cytometry

Noise, in general, is the appearance of nuisance signals in data. Noise is a universal problem in measurement techniques across all science and engineering disciplines. This is no different in flow cytometry, where the molecular properties of individual cells are measured in a sample.

Noise elements in flow cytometry can be categorised in two groups: technical and biological. In the first group are the signals that appear when no sample is running in the cytometer instrument. These low noise signal levels originate from the emission of light, along with its detection by detectors such as photomultiplier tubes (PMTs) or avalanche photodiodes (APDs), as stochastic process [212]. The solution to overcome technical noise

is to perform instrument calibration and apply detection thresholds. Methods include using control particles or material (usually manufactured beads, although these display different size and material properties to biological cells) [213]. Light emitting diode (LED) pulse flashes can also be reliably used to determine the background, signal-to-noise ratio and dynamic range of cytometer PMTs [214]. This ensures only light scatter signals from particles above background are recorded. Assuming calibrations have been performed correctly, technical noise should have a minimal effect on downstream data analysis. On this basis, modelling this source of noise will not be considered further in this study.

Second, are the signals that appear when a biological sample is running. These 'false positive' signals can be attributed to debris, dead cells, doublets (cell aggregates) and other contaminants in the sample. Biological noise events are less easy to filter out during data acquisition, so are recorded to file and removed during downstream data analysis pre-processing steps. Typically, this involves an operator manually applying sequential gates that exclude events based on size, area, fluorescence intensity of viability dye (Figure 6.1). At this point, the data are transformed, and also compensated to remove noise from spillover of fluorescent channels. Automated tools are available to reduce operator variability in these pre-processing and quality control steps [133, 132]. Once the biological noise has been reduced after data pre-processing, analysis can proceed to investigate cell population phenotypes and gain meaningful insights into the data. As well as manual analysis methods, the 'cleaned' data can be fed into dimensionality reduction algorithms that help to visualise the high-dimensional data (e.g. t-SNE, UMAP), and also unsupervised and supervised algorithms that automatically identify cell populations and classify sample types (see Chapter 2).

Despite best practices, noise will remain in flow cytometry datasets. This is evident when comparing real datasets with synthetic ones that artificially model cell populations. As seen in Figure 6.2, the synthetic cell populations appear 'too clean', with outlier events that appear very regular and compact (as a neat halo around the main cluster), and closely follow the probability distribution model (e.g. Gaussian, skew).

In comparison, the outlier events in real data appear more irregular despite rounds of pre-processing to remove debris and to focus on the populations of interest. The pattern of noise does not necessarily closely follow the probability distribution modelled on the data. While still occurring around the vicinity of a main cell population, the outliers appear more randomly distributed and also more spatially spread. The overall plot appearance is more 'messy'.

Hence, to introduce noise into synthetic datasets, it was decided to focus specifically on the outlier characteristic, rather than other noise elements that can be filtered out by

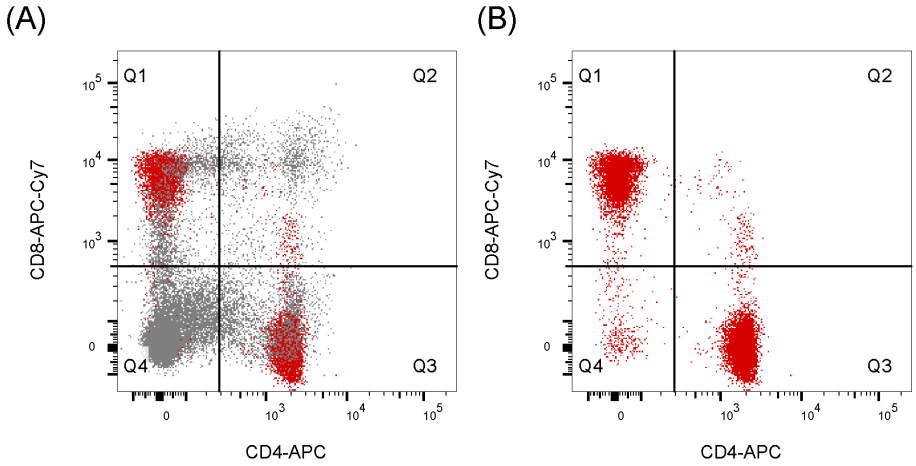


Figure 6.1: The effect of noise on real world data. The CD4 vs CD8 analysis of the data from Figure 1.1 (red dots) is shown (A) highlighted against biological noise events from cell debris, doublets and dead cells that would be present without pre-processing steps to remove them, and (B) with noise events excluded.

gating such as debris, doublets, dead cells and margin events.

6.3 Methods

6.3.1 Synthetic datasets

6.3.1.1 Noise separation dataset

The purpose of the noise separation dataset was to evaluate software performance in the detection of cell populations with increasing levels of noise. This noise-separation dataset builds on the two-cluster separation dataset previously generated in Chapter 4, where distances between clusters ranged along a separation index from well-separated to overlapping. Existing datasets were modified with the injection of a noise layer of 200 points (10% of original dataset), generated using the random uniform distribution function `runif` in R [215], with boundaries defined as the mean ±3 or ±4 standard deviations (SD) of each parameter of the existing data. These two levels of noise are referred to as '3SD' and '4SD' in the remainder of this text. The random noise layer is rectangular in appearance. Dataset properties were:

- Number of clusters: 2
- Cluster size: 1,000 points per cluster
- Separation index (SI) values: from −0.3 to +0.3, at 0.1 intervals

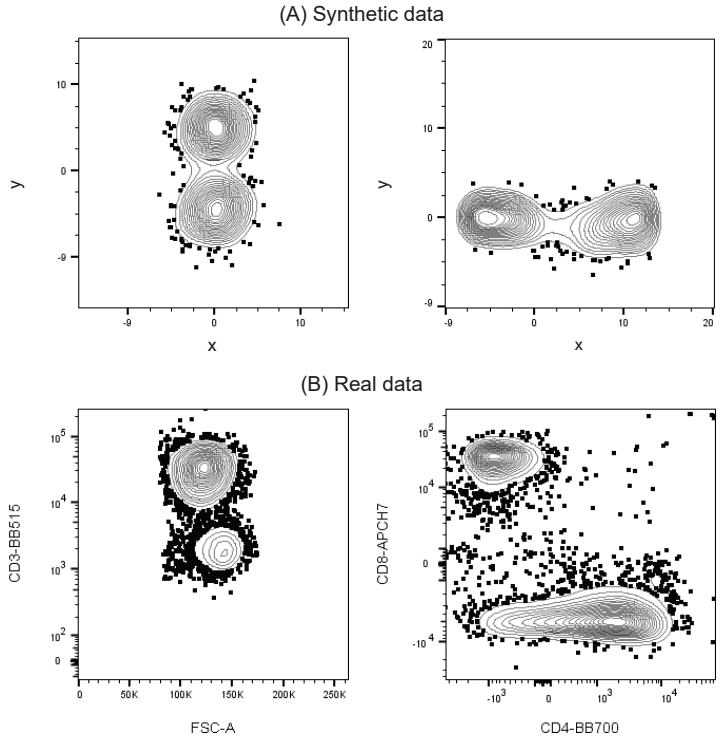


Figure 6.2: Comparison of noise events in exemplar (A) synthetic data against (B) real-world data (PBMCs dataset 2, cf. Section 3.2.8.2) after pre-processing. Data visualised as contour plots with noise emphasised as large dots.

- Noise layer size: 200 points
- Noise distribution: uniform distribution, 3SD or 4SD.

6.3.1.2 Noise skew dataset

The purpose of the noise skew dataset was to evaluate software performance in the detection of skewed cell populations with increasing levels of noise. This dataset extends the skew dataset previously generated in Chapter 4, that contained non-normally distributed clusters. For this noise-skew dataset, clusters with heavy skew ($\alpha = 10$) and facing a tail-to-tail orientation were modified with a new layer of noise elements. The noise layer was introduced to existing datasets as described above in Section 6.3.1.1. Dataset properties were:

- Number of clusters: 2
- Cluster size: 1,000 points per cluster
- α skew value: 10
- Skew orientation: tail-to-tail
- Noise layer size: 200 points

- Noise distribution: uniform distribution, 3SD or 4SD.

6.3.1.3 Noise rare dataset

The purpose of the noise rare dataset was to evaluate software performance in the detection of rare cell populations among increasing levels of noise, and extend the rare cell datasets previously generated in Chapter 5. The rare cluster size of 10 cells was excluded for this noise-rare investigation, because data from Chapter 5 established that none of the software could detect it. The existing datasets were modified with the injection of a noise layer fixed at 200 points, generated using the random uniform distribution with boundaries defined as the range ±1 or ±2 SD of each parameter of the existing data (the range was used rather than the mean because of a lack of noise coverage of the whole dataset when using the mean). These two levels of noise are shortened to '1SD' and '2SD' in the remainder of this text. Clusters displayed normal distributions. Dataset properties were:

- Number of clusters: 2
- Total events: 10^4, 10^5, and 10^6
- Rare cluster size: 50, 100, 500, and 1,000 cells
- Noise layer size: 200 points
- Noise distribution: uniform distribution, 1SD or 2SD.

6.3.2 Software runs

The three noisy synthetic datasets were processed through seven different software: Flock2, flowMeans, FlowSOM, PhenoGraph, SPADE1, SPADE3, and SWIFT. The input parameters used for software runs are listed in Table 6.1. The same software input parameters were used for both the noise-separation and noise-skew datasets, which share similar properties (size of clusters, number of total events).

The noise-rare datasets, which have slightly different properties, were processed with different parameters in Flock2, PhenoGraph, and SPADE3, in keeping with the settings applied for rare cell dataset analysis in Chapter 5. As established previously in Chapter 5, flowMeans and SPADE1 failed to analyse the synthetic rare cell datasets, therefore the decision was made to exclude noise-rare datasets runs on these two software tools because of a lack of comparison to clean dataset outputs. Only runs from the noise-separation and noise-skew datasets were completed for these two software tools. Software outputs were manually interpreted to select the target cluster as previously described.

Table 6.1: User parameters for software runs on noise datasets

Software	Parameter	Dataset		
		Noise separation	**Noise skew**	**Noise rare**
Flock2	Bins	auto	auto	30 (max)
	Density	auto	auto	2 (min)
	Calculate centroids using	Mean fluorescence intensity	Mean fluorescence intensity	Mean fluorescence intensity
flowMeans	Max number of clusters	3	3	Not run
FlowSOM	Number of expected metaclusters	3	3	3
	Grid size	3×3	3×3	3×3
PhenoGraph	*k*, initial clustering	150	150	30
	k, meta-clustering	N/A	N/A	15
SPADE1	Target number of nodes	2	2	Not run
	Downsampled events target	10%	10%	Not run
SPADE3	Outlier density	1st percentile (default)	1st percentile (default)	1st percentile (default)
	Target density	20,000 cells (default)	20,000 cells (default)	100,000 cells
	Number of desired clusters	100 (default)	100 (default)	100 (default)
SWIFT	Input cluster number	2	2	2
	Arcsinh transformation	0	0	0

6.4 Results

Synthetic datasets have been useful toolsets to validate automated algorithms, and previous chapters have shown that a two-clusters synthetic dataset of controlled distances between clusters can be used to assess the cell identification performances of automated software. However, the previous datasets were necessarily limited because of a lack of noise characteristics that did not fully reflect the reality of noisy flow cytometry data, and thus an idealised ability of software to identify cell populations was potentially portrayed. To address this limitation, in this Chapter, synthetic datasets have been developed that include a layer of randomly generated points of uniform distribution to represent the element of noise in real data.

Two factors regarding noise were considered during the design of the datasets. First was the number of noise points to include (either as a fixed value or a percentage of total events), and in this initial study this value was kept constant at 10% of the total number of events in the original dataset. Second, was the distribution of the noise relative to the existing clusters. This was chosen to vary with two levels: noise limits defined by the mean of existing data ±3SD, or a wider distribution at ±4SD. Figure 6.3A illustrates the noise layer generated over the existing two clusters. Note that the two clusters are symmetrical ellipses with normal distributions along both x and y axes, with slightly random shape attributes as described in Chapter 3.

6.4.1 Results of noise-separation dataset runs

The noise datasets were processed through different software, and for each run, the cell count output of cluster 1 was compared with the reference of 1,000 points. A caveat of this analysis was that the additional 200 noise points in the dataset increased its total events to 2,200 points, so if a software bisected the data, 1,100 points per cluster would be returned. To account for this, two 'target' values including/excluding noise events were used for comparison purposes.

The results found considerable variation in the ability of different software to identify cell populations as the levels of noise increased from 3SD to 4SD. The effect of noise ranged from almost no impact on performance in SPADE3 across the range of SI values tested (Figure 6.3B), to significant deterioration in performance in FlowSOM, where even well-separated clusters with SI ≥ 0 could not be detected (Figure 6.3C).

As expected, the introduction of noise caused a reduction in performance compared with the noiseless separation datasets. This was observed for all software, and across the

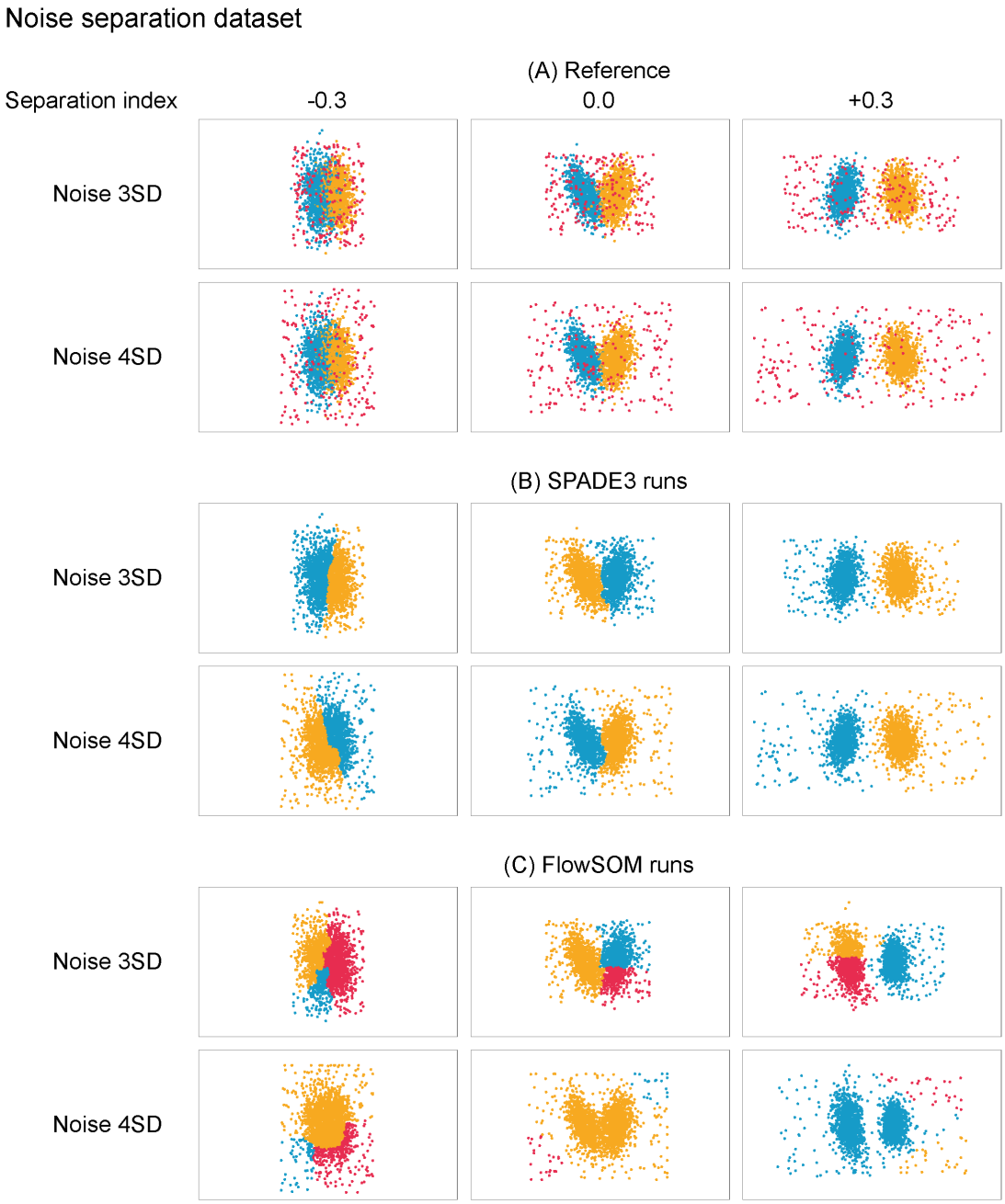


Figure 6.3: Clustering examples from the noise-separation dataset. (A) Reference dataset, with two clusters and the additional noise layer illustrated. (B) SPADE3 performance appeared to be unaffected by the increasing level of noise, with similar clustering characteristics observed between noise at 3SD and 4SD. (C) FlowSOM clustering performance deteriorated as noise levels increased from 3SD to 4SD, even as clusters became well-separated at separation index values of 0 and above.

whole range of SI values tested (Figure 6.4).

Interestingly, the change in noise distribution from 3SD to 4SD revealed different trends in performance from different software (Figure 6.4). For certain software, accuracy and repeatability appeared to worsen as the noise level changed from 3SD to 4SD. For example, at SI of -0.1, the FlowSOM difference to reference count widened from 168 ± 79 to 735 ± 487, and SWIFT difference to reference count also increased from 272 ± 182 to 439 ± 274.

The opposite trend in performance was observed for other software, where the increase from 3SD to 4SD appeared to improve accuracy and repeatability (Figure 6.4). This was observed in SPADE1, where for example at SI of -0.1, the difference to reference count narrowed from 363 ± 273 to 258 ± 171. Similarly, with Flock2, at SI of $+0.1$ the outputs improved from 139 ± 78 to 110 ± 27 as the noise level moved from 3SD to 4SD. This trend in Flock2 was less apparent as clusters merged below a SI value of 0. This finding is possibly related to the density of noise, in which the sparser distribution at the 4SD level makes the main cluster easier to detect.

For PhenoGraph and SPADE3, no significant differences in performance were observed between the different levels of noise.

SWIFT was the only software that saw its difference to reference outputs fall below the 'target including noise events' of 100 (see Figure 6.4, SWIFT SI $\geq +0.2$); all other software did not exclude noise events in their analyses. These results suggest that the SWIFT algorithm was able to filter out noise from datasets with well-separated clusters, possibly as a function of discarding events that do not fit a Gaussian distribution.

Further insight into the variation in software performances was carried out by calculating the coefficient of variation (CV) of the software outputs for each SI value and at each noise level (Figure 6.5). This analysis showed highly similar trends in variation along the SI range at all three levels of noise for Flock2, PhenoGraph, SPADE1 and SPADE3. This suggests the injection of noise in the separation dataset had no significant impact on repeatability among these software tools. In comparison, both flowMeans and SWIFT displayed mostly higher CVs at each SI value as the noise levels increased. For FlowSOM, while trends in variation were similar at the noiseless and 3SD noise levels, there was a large increase in the CVs of outputs from SI ≥ -0.2 at the 4SD noise level.

The CVs across all SI values was calculated (Figure 6.6). This grouped analysis revealed that the mean CV was lower on clean datasets (with no noise) compared to noisy datasets for all software runs except SPADE1 and SPADE3. Comparisons of the mean CVs between the 3SD and 4SD noise levels showed all software had higher CVs at 4SD, except SPADE3. The software giving the least variation was PhenoGraph for the

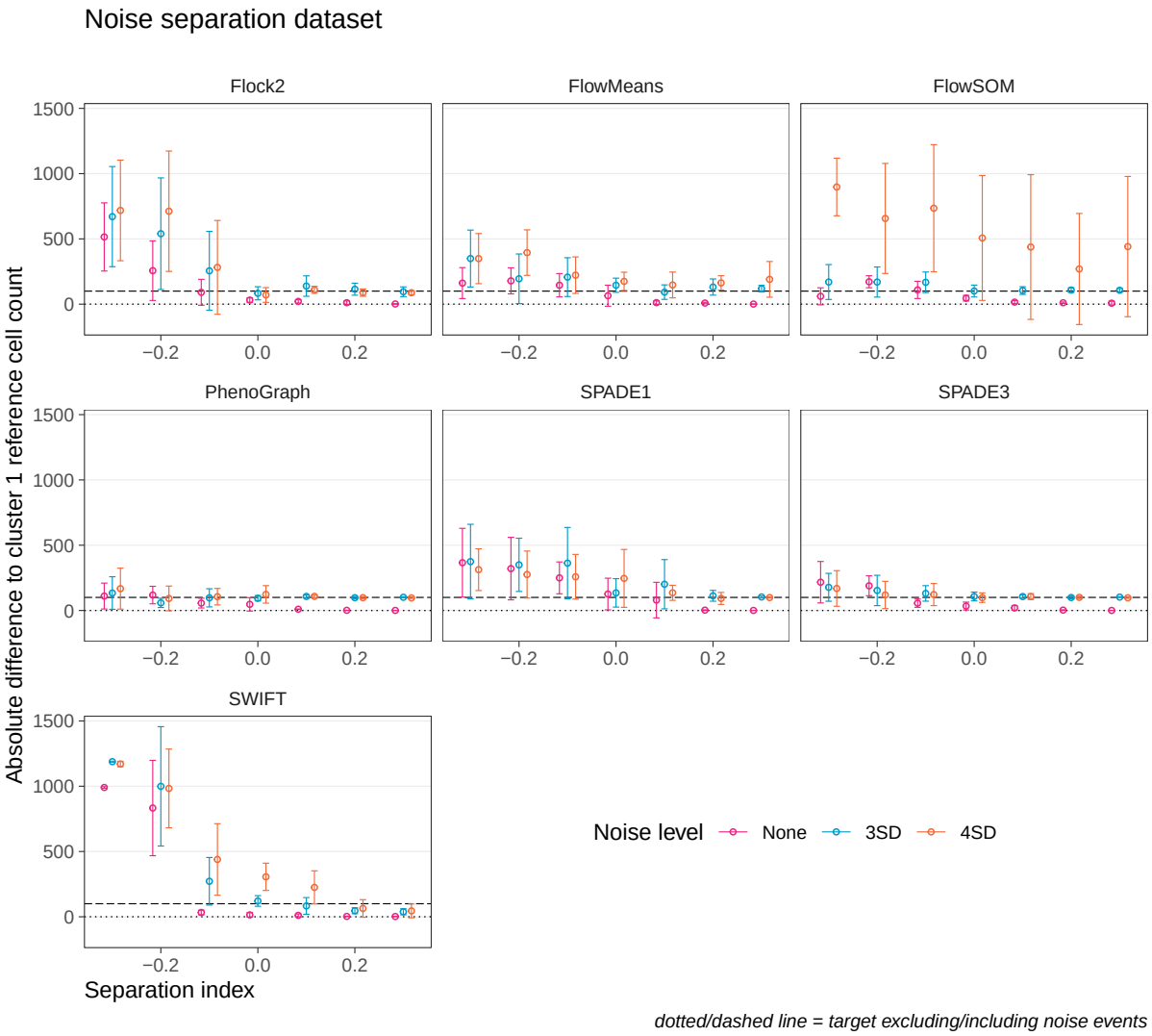


Figure 6.4: Performance of different software on a two-cluster separation dataset with noise elements. Comparison with datasets containing no noise show that all software performances worsened with the addition of noise, across all SI values. The deterioration in FlowSOM performance from noise level of 3SD to 4SD is apparent, compared with other software such as PhenoGraph and SPADE3 that are less affected by noise.

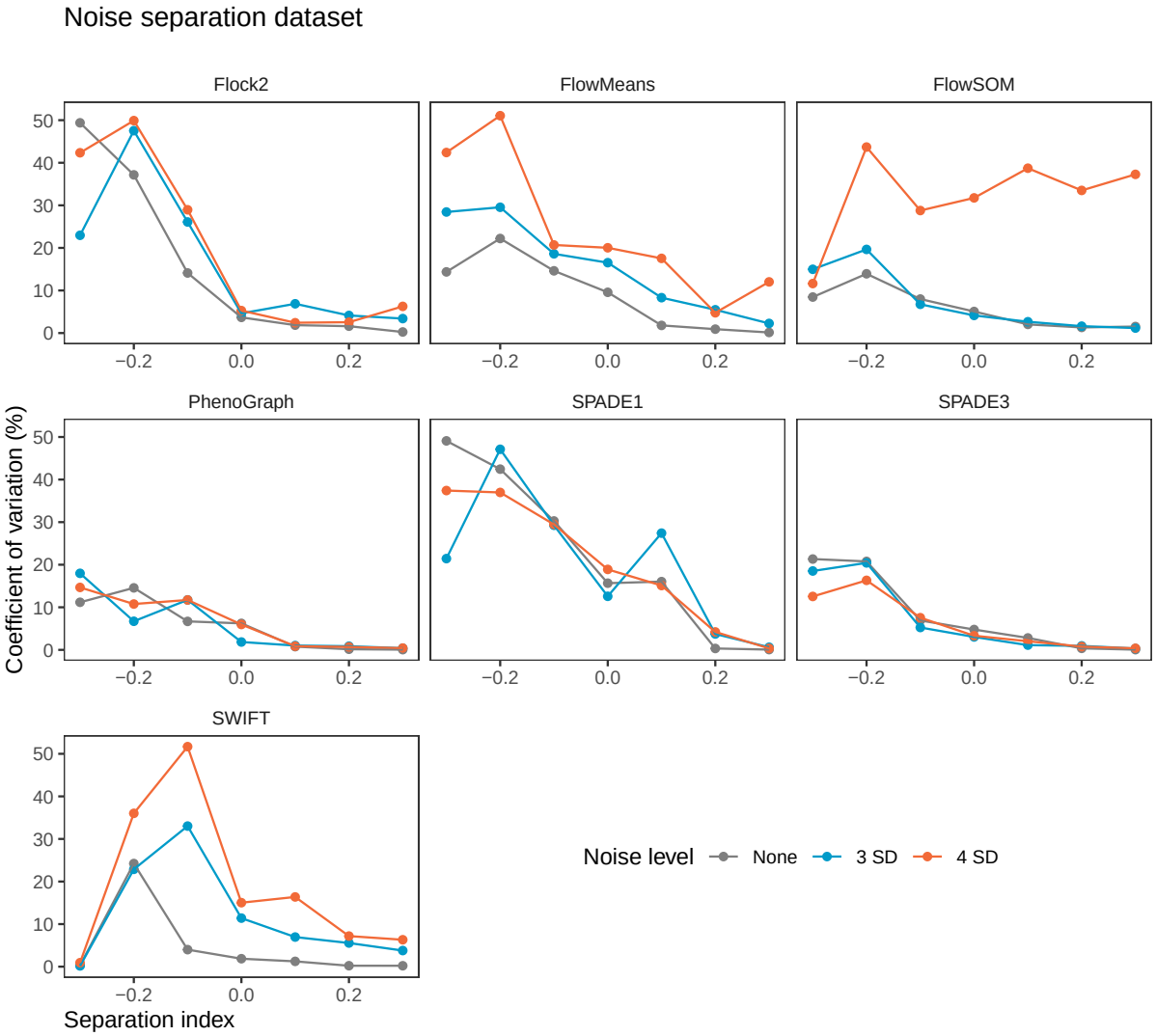


Figure 6.5: Coefficient of variations of software outputs at each SI value, analysing datasets with increasing levels of noise.

3SD noise level and SPADE3 for the 4SD noise level (CVs of 5.8% and 6.1%, respectively), and the most variation was observed for SPADE1 in the 3SD condition, and FlowSOM in the 4SD condition (CVs of 20.3% and 32.2%, respectively). Of note, FlowSOM analysis of the 4SD noise dataset showed a significantly higher CV (32.2%) compared to both the no noise (5.8%) and 3SD (7.3%) noise conditions ($p < 0.001$, one-way ANOVA with Tukey multiple comparison test).

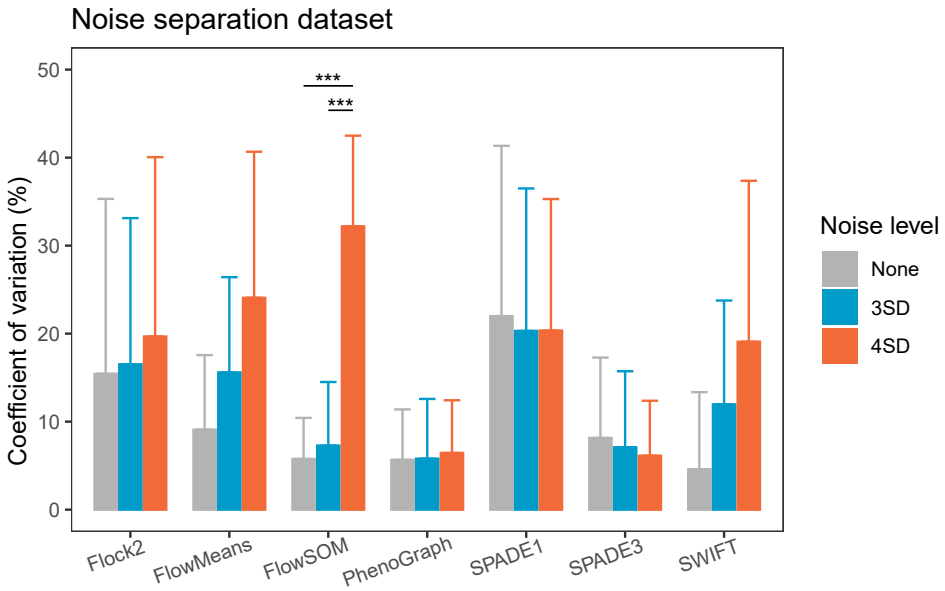


Figure 6.6: Comparison of coefficient of variations across all SIs, between different software analysing datasets with increasing levels of noise. ***$p < 0.001$ (One-way ANOVA).

6.4.2 Results of noise-skew dataset runs

The next question to explore was the effect of noise on software detection of clusters with skewed distributions. Previous data in Chapter 4 showed SWIFT performance was the most affected by skew data. In contrast, FlowSOM and SPADE3 were the least affected, and Flock2 appeared to improve performance with increasing skew.

The noise-skew dataset generated here focused on well-separated clusters with a SI of +0.1 in a tail-to-tail orientation. As with the previous normally distributed datasets, the original skew datasets created in Chapter 4 were modified through addition of noise distribution levels of 3SD and 4SD. The noise-skew dataset designs are illustrated in Figure 6.7A.

Software issues encountered during attempts to run the two-cluster skew dataset (with no noise) through SPADE1 on the web-based Cytobank platform meant outputs from this

condition could not be obtained, albeit there were no problems with running the datasets with additional noise elements, so these successful SPADE1 outputs have been included in the analyses below.

Results of the clustering from software here showed similar characteristics to those from the normally distributed clusters: FlowSOM was not able to detect the skewed cluster among 4SD levels of noise (Figure 6.7B), while SPADE3 displayed similar clustering patterns at both 3SD and 4SD levels and appeared relatively unaffected by the noise (Figure 6.7C). SWIFT again discarded events that did not fit a Gaussian distribution, which, as well as events on the boundaries, also included events at the tails of the main clusters, here located in between the two clusters (Figure 6.7D).

In a similar manner to the dataset with normal clusters, addition of noise saw a decline in performance from all software (Figure 6.8). For five of the software tested on the noise-skew dataset, the cell population detection performance declined as the noise level changed from 3SD to 4SD. FlowSOM results worsened from 80 ± 54 to 496 ± 527 in a comparable pattern to its response to noise levels in the noise-normal dataset (noting large error bars in Figure 6.8). Outputs from Flock2, flowMeans, SPADE1 and SWIFT also declined as the noise level shifted from 3SD to 4SD, but to a lesser extent (Figure 6.8, bottom). This trend seen in Flock2 and SPADE1 deviated from the findings with the noise-normal datasets, where accuracy and repeatability appeared to improve from 3SD to 4SD noise levels, and thus is potentially a response to the skewed distributions. No significant differences in performance were found between the 3SD and 4SD levels of noise for PhenoGraph (105 ± 34 to 100 ± 34) and SPADE3 (87 ± 42 to 90 ± 34).

Analysis of CVs from the noise-skew dataset (Figure 6.9) showed PhenoGraph as the most consistent software, with lowest CVs for both 3SD and 4SD conditions (3.0% and 3.1% respectively) compared with other software tested. SWIFT reported the highest CV of 18.6% for the 3SD condition, while FlowSOM reported the highest CV of 35.2% for the 4SD noise level, which was also significantly higher than its CVs for the no noise and 3SD noise levels (4.3% and 5.4%, respectively). On the basis of this metric, Flock2, FlowSOM and flowMeans gave higher variability when processing noise at 4SD compared with 3SD levels. In contrast, SPADE1, SPADE3 and SWIFT showed higher variability when processing noise at 3SD levels, thus suggesting better performance at the higher 4SD noise level.

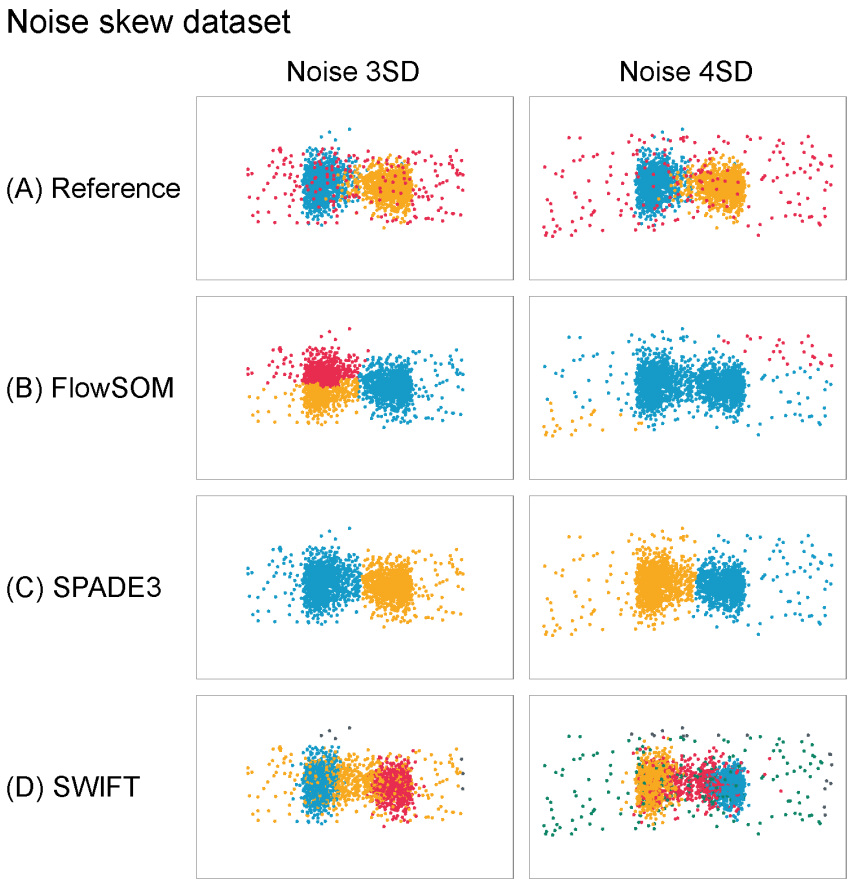


Figure 6.7: Clustering examples from selected best and worst performing software on a two-cluster skew dataset (skew $\alpha = 10$; tail-to-tail orientation; SI $= 0.1$) with noise elements.

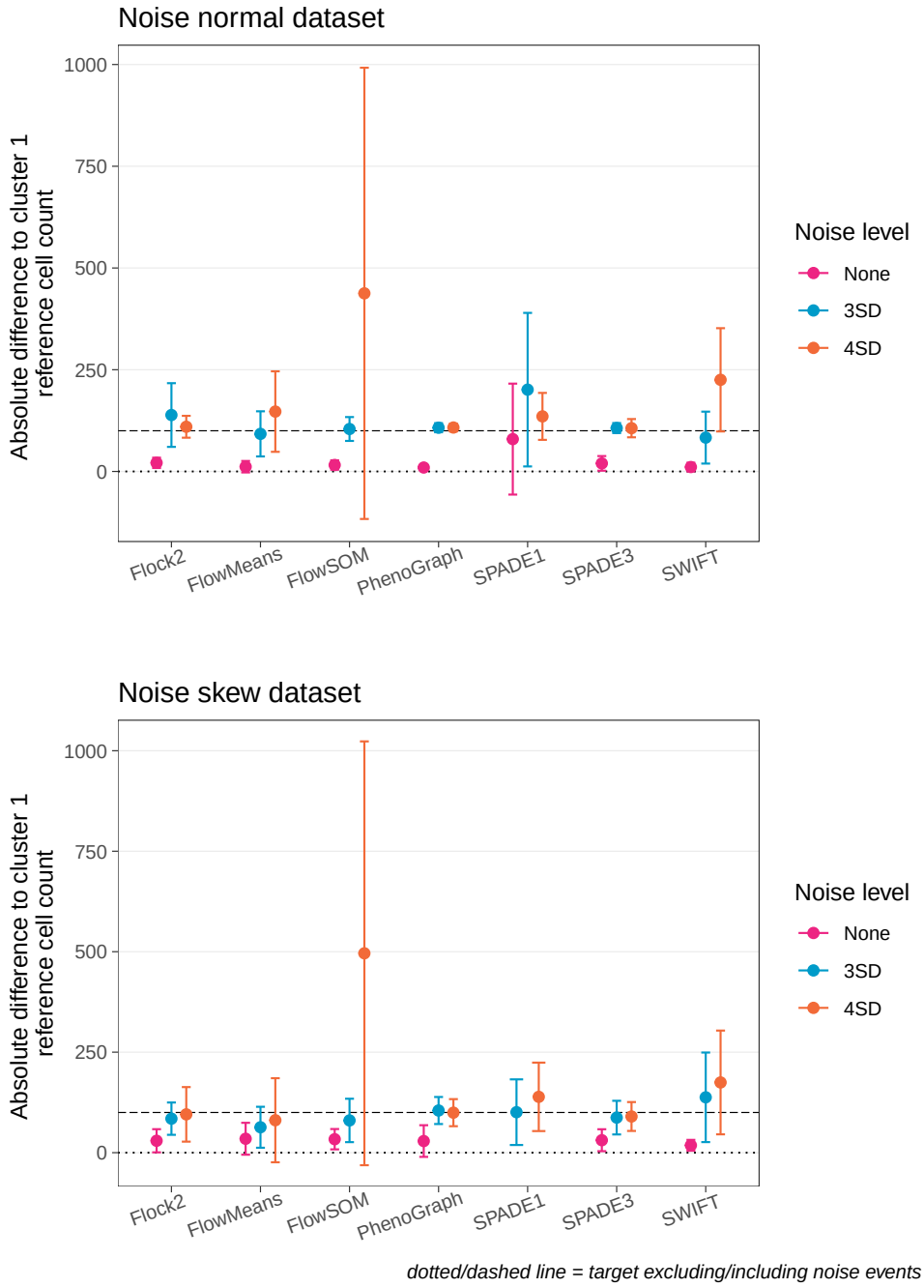


Figure 6.8: Performance of different software on a two-cluster normal dataset (top) compared with a skew dataset (bottom) with noise elements. All clusters with SI = 0.1.

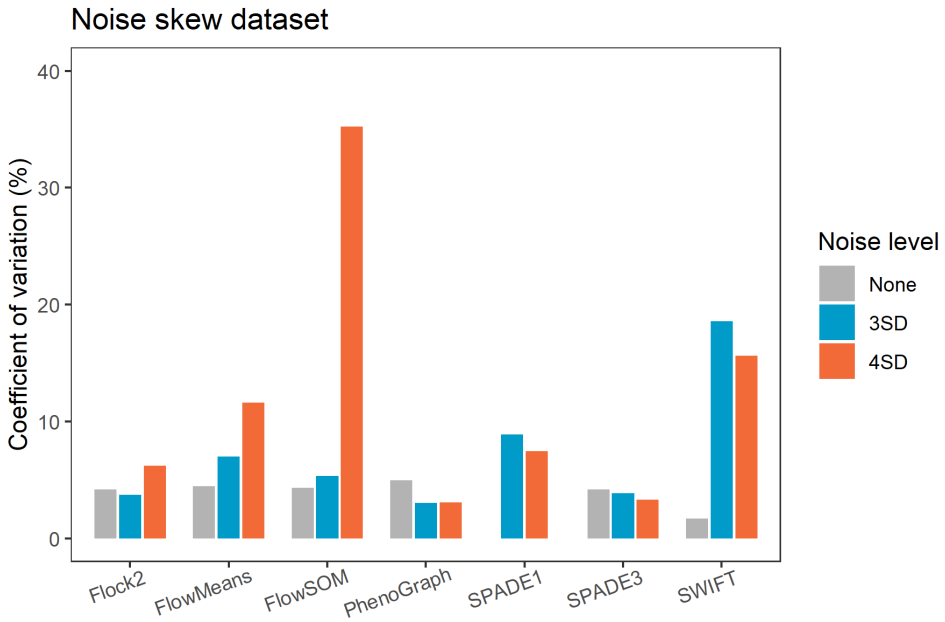


Figure 6.9: Comparison of coefficient of variations between different software analysing two-cluster skew datasets with increasing levels of noise.

6.4.3 Results of noise-rare dataset runs

The effect of noise on software performance on the detection of rare cell populations was next investigated. Previous work with synthetic flow cytometry datasets containing rare populations showed that automated data analysis software have different performance behaviours (Chapter 5). SWIFT and FlowSOM could detect a lower absolute number of cells, followed by Flock2. In contrast, SPADE3 and PhenoGraph gave poor performances, being unable to identify rare populations with accuracy and repeatability.

The addition of noise was expected to result in a loss of performance for rare cluster identification, due to a weaker signal-to-noise ratio in the presence of noise that may cause the rare cluster to become less sensitive to detection.

To address the effect of dataset size on rare population frequencies, rare-noise synthetic datasets with 10^4, 10^5 and 10^6 total events were processed through the different software tools. The results presented here compare the various software's ability to detect the rare population among synthetic datasets with different levels of noise elements.

6.4.3.1 Noise-rare 10^4 dataset

Designs of the noise-rare 10^4 dataset are illustrated in Figure 6.10A, alongside examples of clustering outputs from the best performing software of Flock2, FlowSOM and SWIFT (Figure 6.10B, C, and D, respectively).

Flock2 managed to isolate the 50-cell rare population with counts of 108 ± 20 and 116 ± 13 at the 1SD and 2SD noise levels, respectively (Figure 6.11). There was a suggestion that the sparser distribution of noise events at the 2SD level allowed Flock2 to detect the rare cluster with better accuracy and repeatability — as observed at 100-cell level, where the gap to the target narrowed by ∼50 cells, from 170 ± 17 at the 1SD level to 118 ± 90 at the 2SD noise level. The rare cluster became increasingly identifiable to Flock2 as its size increased to 500 and 1000 cells, with outputs falling within ±20% of the target count (Figure 6.10B).

FlowSOM was previously able to detect the rare cluster of 50 cells in the non-noise dataset. As expected, the injection of noise reduced its accuracy and repeatability, but nevertheless its outputs were comparable or closer to target than Flock2, with counts at the 50-cell level of 83 ± 48 and 88 ± 28 for 1SD and 2SD noise levels, respectively. At 500 cells, FlowSOM reported values within ±20% of the target count, of 532 ± 93 and 580 ± 41 for 1SD and 2SD noise levels, respectively. However, FlowSOM failed to successfully separate the clusters at the 1,000 rare cell level, with the partition occurring horizontally across the tops of the two clusters (Figure 6.10C). This clustering behaviour

from FlowSOM resembled earlier observations in its runs of the noise-separation and noise-skew datasets.

SWIFT also previously achieved a limit of detection (LoD) of 50 in 10^4 cells. Here, with the noise-rare dataset at rare clusters of 100 and below, SWIFT almost appeared to be able to remove the entire noise layer applied in the synthetic datasets, however it also discarded the rare clusters alongside (Figure 6.10D). SWIFT was able to detect the 500-cell cluster among noise with counts of 511 ± 139 for the 1SD noise level, however it reported an underestimate >100 cells at the 2SD level with 369 ± 55.

SPADE3 limits of detection in the noise-rare datasets generally matched those of the non-noise dataset, with output counts of the 1,000 rare cell cluster of $1,094 \pm 4$ and $1,103 \pm 13$ for 1SD and 2SD noise levels, respectively. Of note, the rare cluster of 500 cells in the 1SD noise level appeared to be detectable by SPADE3 with a reported count of 590 ± 7, its previous presence being undetected in the noiseless datasets (Figure 6.11).

PhenoGraph remained unable to detect the rare cluster in the dataset with noise added, with outputs that were orders of magnitude away from the target count.

6.4.3.2 Noise-rare 10^5 dataset

By increasing the total events in the noise-rare dataset to 10^5, the ratio of rare cells to the major population widened while the ratio of rare cells to noise (fixed at 200 events) remained the same as the 10^4 dataset. This dataset (Figure 6.12A) enabled the investigation into the effect of noise on software detection of rare populations at lower frequencies. The results presented here once again show a varied response by different software to the addition of noise in a rare dataset (Figure 6.13).

Increasing the noise level from 1SD to 2SD did not appear to affect the clustering patterns of Flock2 (Figure 6.12B). Flock2 runs on the noise datasets matched the LoD given with non-noise dataset at 1,000 cells (1%). The counts given at the 500-cell level for the 1SD and 2SD noise datasets were of improved accuracy and repeatability compared with the non-noise dataset, where the outputs were an order of magnitude away from target (Figure 6.13).

FlowSOM was observed to be the best performing software on this dataset, with its response to the additional noise elements not dissimilar to what was seen in the 10^4 dataset. As with Flock2, increasing the noise levels from 1SD to 2SD did not appear to have a large impact on clustering behaviours (Figure 6.12C). At the 50-cell level, FlowSOM reported rare counts of 86 ± 31 and 140 ± 55 for 1SD and 2SD noise levels, respectively. Likewise at the 100-cells level with counts of 132 ± 11 and 161 ± 64. In the

same manner as the 10^4 noise-rare dataset, FlowSOM achieved rare cell detection within $\pm 20\%$ of the target count at 500 in 10^5 cells (0.5%) for both noise levels.

The impact of noise on the ability of SWIFT to detect the rare cluster was relatively minor (Figure 6.12D). The limit of detection at no noise was 500 cells. At the equivalent rare cell level, SWIFT reported counts of 588 ± 6 with 1SD level of noise, and again as with the 10^4 dataset under-reported the 2SD noise level by >100 cells with a count of 368 ± 114.

Similarly to the runs on rare dataset with no noise, PhenoGraph and SPADE3 remained unable to detect the rare population for all conditions in the noisy datasets (Figure 6.13).

6.4.3.3 Noise-rare 10^6 dataset

The designs of the two-cluster rare 10^6 datasets with 1SD and 2SD levels of noise are depicted in Figure 6.14A). This noise-rare dataset was processed through software for a complete comparison to the non-noise rare datasets from Chapter 5.

Once again, there was little difference in the general clustering outputs between 1SD and 2SD noise levels for Flock2 (Figure 6.14B). However, when comparing performance to the datasets with no noise, Flock2 appeared to perform better when noise is added (Figure 6.15). For example, at the 500 rare cells level, while the rare population remained undetected for Flock2 in the datasets without noise, with 1SD noise, Flock reported a count of 621 ± 97. Likewise for the 2SD noise level Flock2 was able to detect the 1,000-cell cluster with reasonable accuracy and repeatability $(1,107 \pm 19)$.

Noise elements in this dataset also appeared to facilitate FlowSOM detection of rare cells (Figure 6.15). Taking into account the extra 200 noise events, FlowSOM came closest to detection of 50 cells among the other software tested here (counts of 133 ± 32 and 115 ± 12 for 1SD and 2SD noise levels, respectively) (Figure 6.14C). Likewise at the 100 rare cell level, FlowSOM counts for the 1SD and 2SD noise levels (145 ± 5 and 150 ± 7, respectively) were an improvement to the non-noise dataset (not detected).

The addition of noise in this dataset saw SWIFT unable to detect the rare cluster (Figure 6.14D), with counts given that were orders of magnitude above the target value. Finally, in line with results from the 10^5 dataset, PhenoGraph and SPADE3 failed to detect the rare cluster for all conditions presented here at 10^6 total events.

Taken together, findings from the noise-rare datasets show that software ability to detect rare populations generally deteriorate as noise elements are injected into the data. Using an example threshold of true rare cell count $\pm 20\%$, the LoD at 10^4 total events is

raised from 50 cells (0.5%) with no noise to 500 cells (5%) with 1SD noise for Flock2, FlowSOM and SWIFT (Table 6.2). Note that for all the noise-rare dataset runs, flowMeans and SPADE1 results were absent because of technical issues with software that led to failed runs, as mentioned in Chapter 5.

The impact of increasing the bounds of the noise distributions from 1SD to 2SD was slightly different for different software. For instance at 10^4 total events, while SWIFT LoD worsened from 5% to 10%, Flock2 LoD improved from 5% to 1%. Along these lines, the injection of noise at 10^6 total events appeared to have enabled rare population detection for Flock2.

Table 6.2: Limits of detection (LoD) of software processing rare cell dataset with increasing levels of noise. flowMeans and SPADE1 excluded because of failed runs.

		Noise level		
Total events	**Software**	**None**	**1SD**	**2SD**
10^4	Flock2	50 (0.5%)	500 (5%)	100 (1%)
	FlowSOM	50 (0.5%)	500 (5%)	500 (5%)
	PhenoGraph	Not detected	Not detected	Not detected
	SPADE3	1000 (10%)	500 (5%)	1000 (10%)
	SWIFT	50 (0.5%)	500 (5%)	1000 (10%)
10^5	Flock2	1000 (1%)	1000 (1%)	1000 (1%)
	FlowSOM	100 (0.1%)	500 (0.5%)	500 (0.5%)
	PhenoGraph	Not detected	Not detected	Not detected
	SPADE3	Not detected	Not detected	Not detected
	SWIFT	500 (0.5%)	500 (0.5%)	1000 (1%)
10^6	Flock2	Not detected	1000 (0.1%)	1000 (0.1%)
	FlowSOM	500 (0.05%)	500 (0.05%)	500 (0.05%)
	PhenoGraph	Not detected	Not detected	Not detected
	SPADE3	Not detected	Not detected	Not detected
	SWIFT	500 (0.05%)	Not detected	Not detected

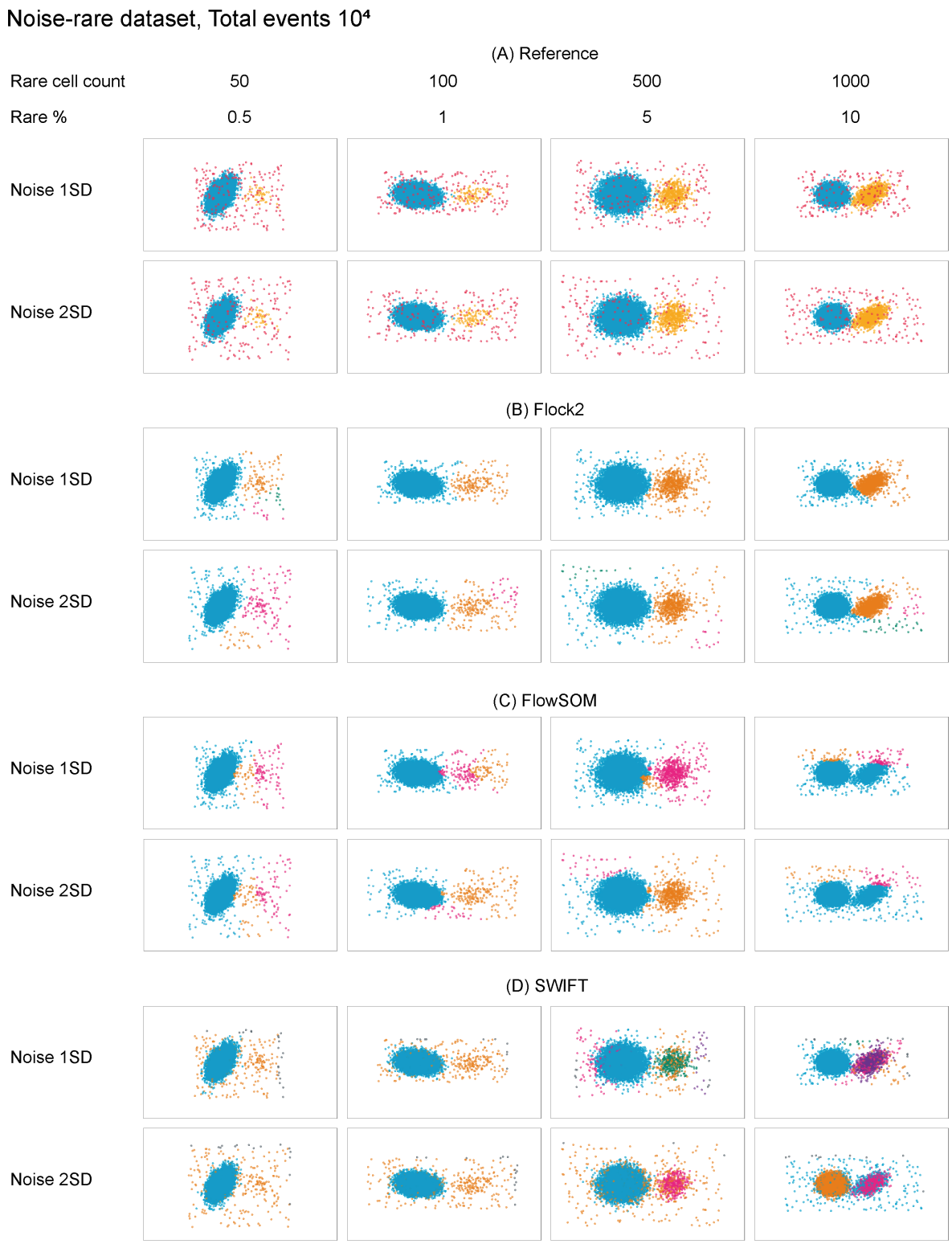


Figure 6.10: Clustering examples from selected software runs on a two-cluster rare dataset with 10^4 total events, with 1SD and 2SD levels of noise.

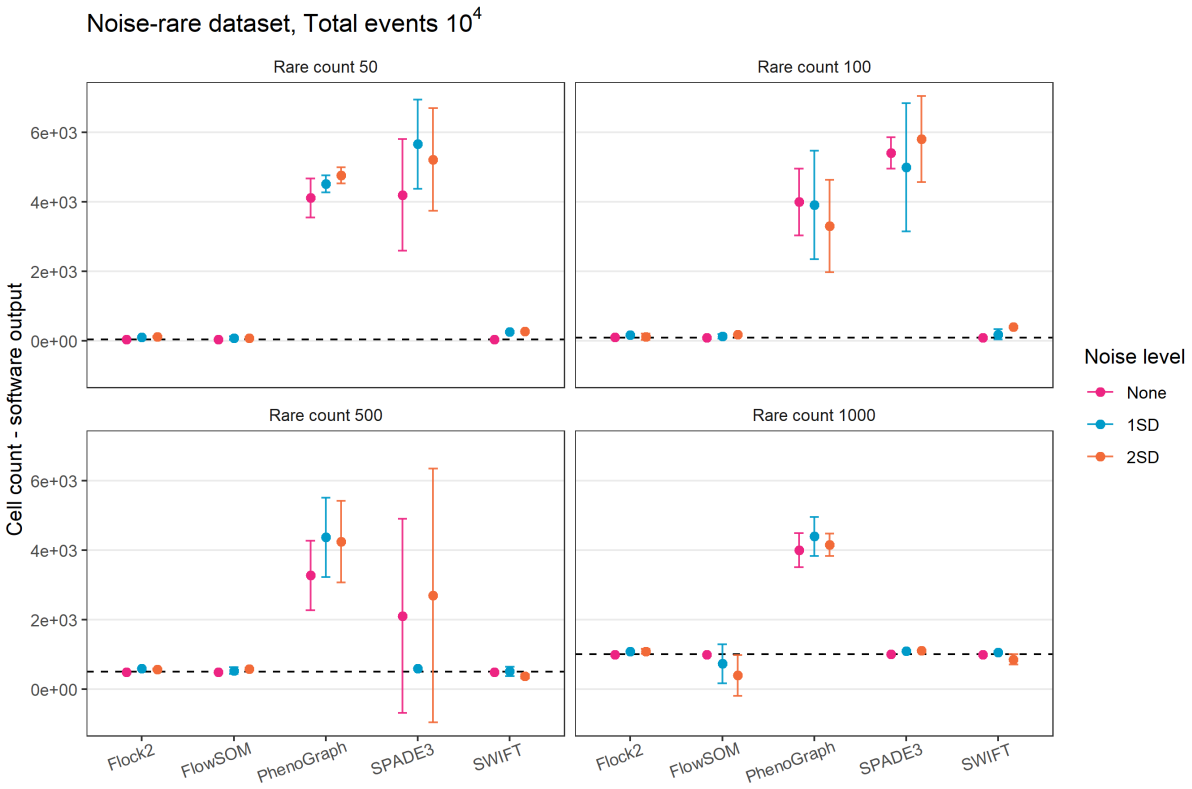


Figure 6.11: Performance of different software on a two-cluster rare dataset with 10^4 total events, with noise elements.

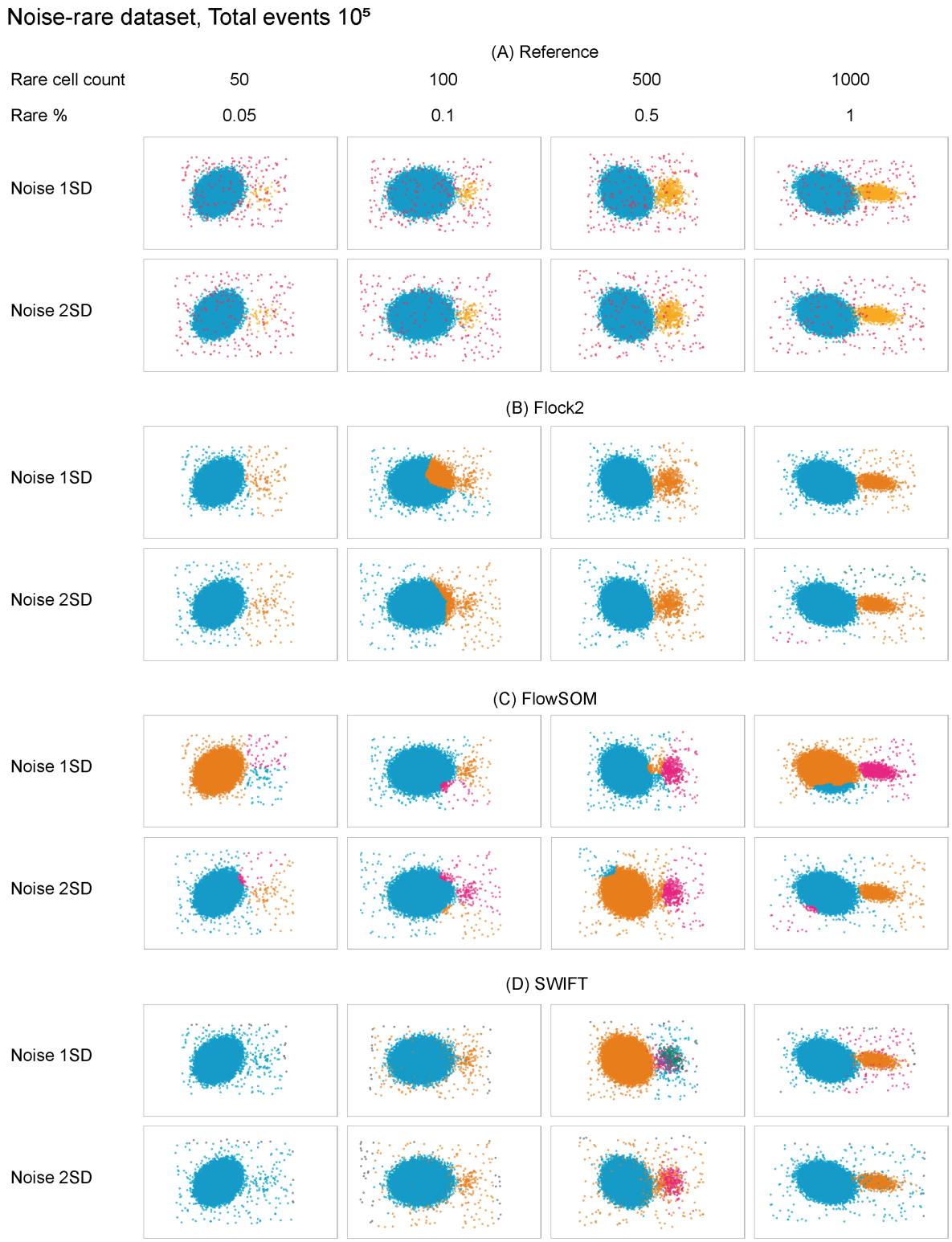


Figure 6.12: Clustering examples from selected software runs on a two-cluster rare dataset with 10^5 total events, with 1SD and 2SD levels of noise.

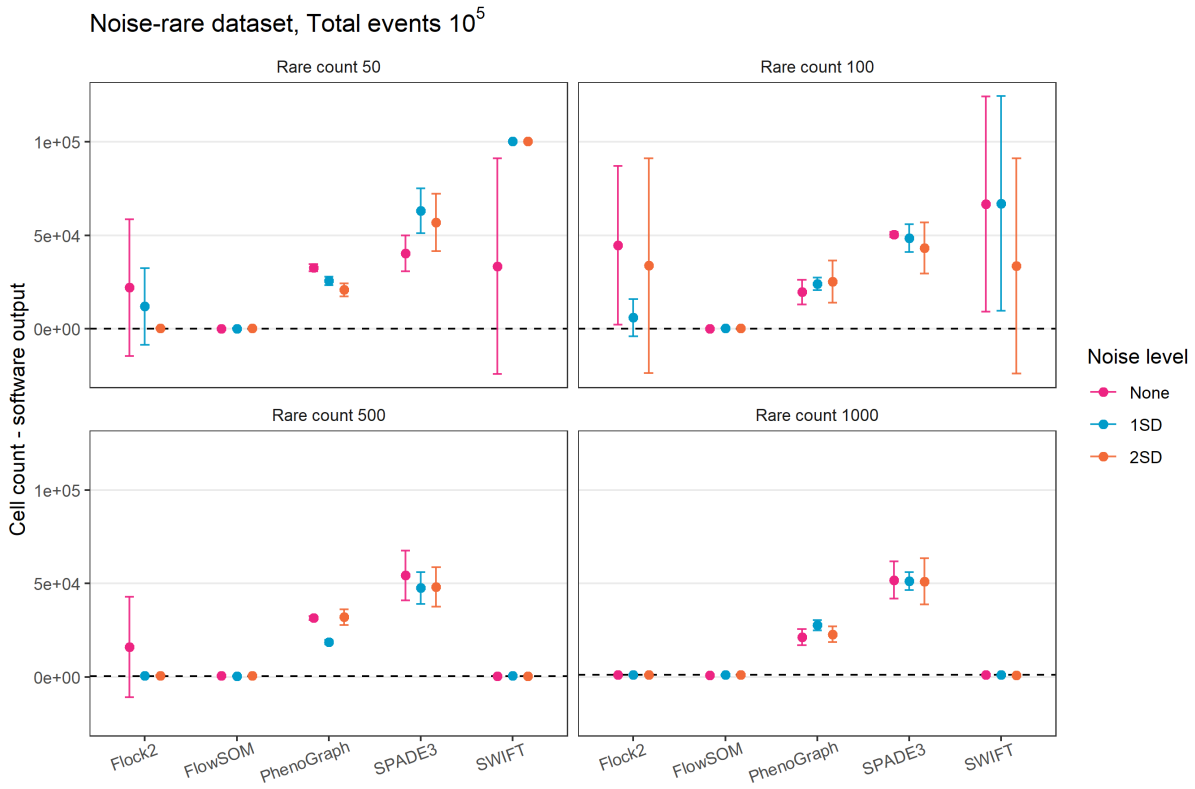


Figure 6.13: Performance of different software on a two-cluster rare dataset with 10^5 total events, with noise elements.

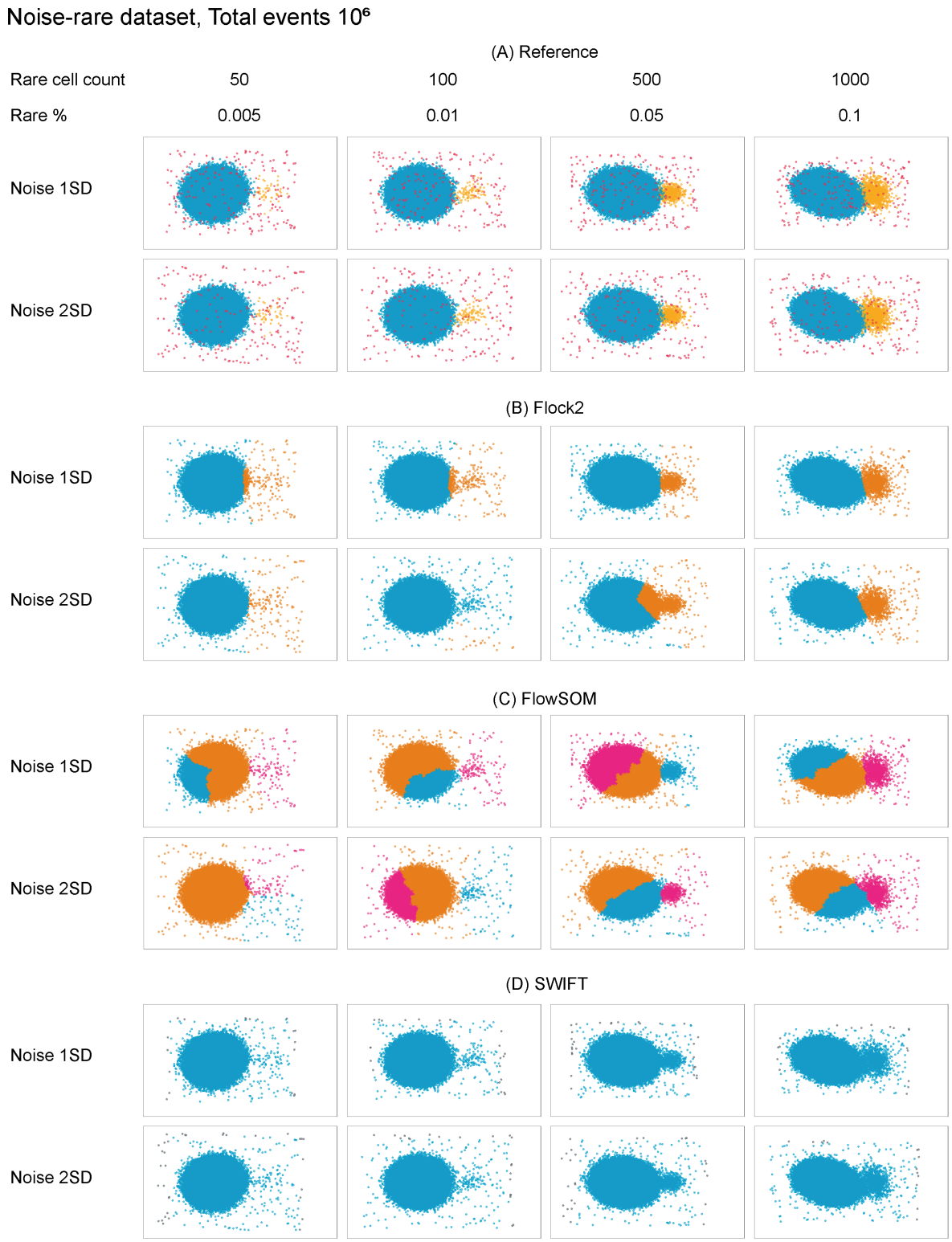


Figure 6.14: Clustering examples from selected software runs on a two-cluster rare dataset with 10^6 total events, with 1SD and 2SD levels of noise.

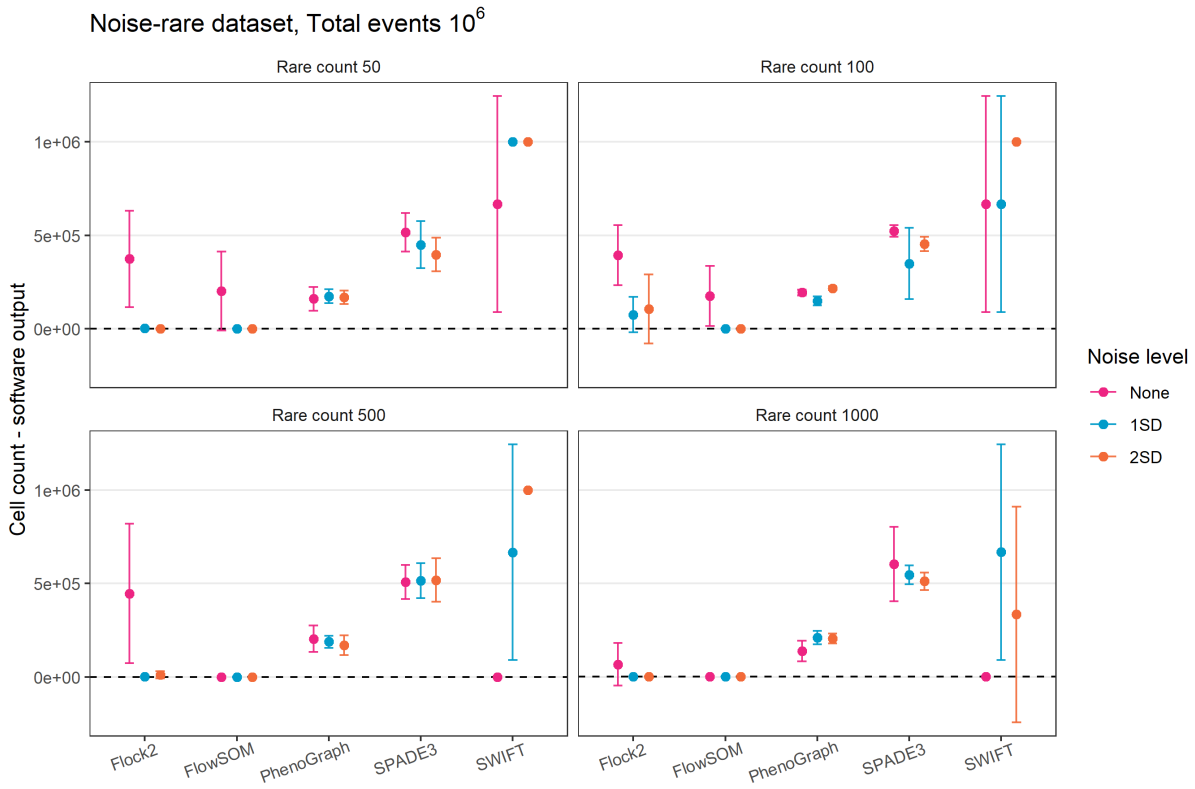


Figure 6.15: Performance of different software on a two-cluster rare dataset with 10^6 total events, with noise elements.

6.5 Discussion

This Chapter has demonstrated the design and generation of artificial noise in synthetic flow cytometry datasets, specifically noise applied with a random uniform distribution with variable range limits, to simulate outlier properties of cell populations that have already undergone pre-processing steps.

The noisy datasets were processed through seven automated software. As expected, the addition of noise saw a decline in accuracy and repeatability of cell population identification from all software tested. This study also found different software behaviours in response to broadening noise limits, which most likely can be attributed to the different clustering strategies that each one employs.

Comparisons of software runs on separation and skew noiseless and noisy datasets revealed that certain software were more sensitive to noise than others. Outputs from PhenoGraph and SPADE3 were robust, with similar clustering characteristics and no significant differences in performance across the different levels of noise tested. This is possibly because of the overclustering techniques used by both, along with downsampling in SPADE3 that reduces the outlier effects, and the manual interpretation of output clusters in PhenoGraph that gives an artificial boost to its performance.

Regarding the possible impact of subsampling (or downsampling, in the case of SPADE3), SWIFT performs an analogous random 'weighted iterative sampling' step during its Gaussian mixture model based clustering to better resolve smaller subpopulations. However, since SWIFT was not as robust to noise levels increasing from 3SD to 4SD as SPADE3, the question remains as to how effective separate subsampling methods are (as an individual step within the entire clustering algorithm) at removing noise.

The subsampling picture is further complicated when considering two algorithms that do not apply subsampling steps in their respective algorithms: Flock2 and FlowSOM. While the former (density-based clustering) saw improvements in performance from 3SD to 4SD noise levels for well-separated clusters, the latter (self-organising map) saw the largest deterioration in performance between 3SD to 4SD noise levels, and despite clusters being well-separated, FlowSOM was unable to meaningfully partition the data in the 4SD noise levels.

The results from software runs on the noise-rare datasets showed that limits of detection of the rare cell population worsened compared with the noiseless datasets, with rare event sizes of 50 and 100 cells beyond the limits of detection of most software. This was to be expected because the layer of noise interfered with the already weak signal from the rare cells. However, it was interesting to see certain software, in particular Flock2,

perform better when noise is added. A possible reason for this improvement may be that the additional noise layer expanded the (two-dimensional) space that data points existed in, which then changed the way the grid-based density clustering algorithm was implemented by Flock2; events in the rare cluster now potentially appeared in a data-dense gridded region, alongside other gridded regions that only contained noise, which in turn affected how dense regions were merged together in the clustering algorithm. Previously, in rare datasets without noise, the rare cluster was merged with the main cluster, and the final clustering bisected the main cluster to return two clusters (Chapter 5). Here, it would appear that the rare cluster instead merged with neighbouring noise events, and the merged group was recognised as a separate population from the bulk cluster.

The findings presented here will give users valuable insight into the robustness of software to noise elements in their data, leading to inform decisions on the appropriate level of pre-processing steps required to 'clean' their data before input into algorithms.

This study leaves room for further work to optimise the noise layer, for instance by varying the size or proportion of noise to the main data, or applying other models and distributions of outliers in addition to the uniform distribution used here. Comparison with a real-world dataset with different levels of noise may be necessary, however obtaining real datasets with controlled cell population outlier specifications may be difficult. Furthermore, the toolsets used here to increase the complexity of synthetic datasets would contribute towards developing high-quality artificial models that closely mimic real-world examples.

6.6 Chapter conclusions

- A range of noise levels have been modelled into synthetic flow cytometry datasets.
- The synthetic noise datasets have been processed through different software.
- FlowSOM performance was found to deteriorate with the introduction of noise.
- PhenoGraph and SPADE3 performances were less affected by noise.
- Runs on noise-rare dataset revealed the rare event sizes of 50 and 100 cells were beyond the limits of detection for most software.
- Findings from these experiments will have implications on the pre-processing steps that users apply in the analysis of their own data.

Chapter 7

NEQAS

7.1 Introduction

The work covered in previous chapters has focussed on comparison of cell population identification performance between different software, over a range of synthetic datasets with controlled properties, and with real datasets. The variability between software has been extensively explored, however a gap exists in the comparison of variation between software data analysis against human participants. As such, it was important to understand the difference in variation between analysis performed by a range of available algorithms versus the variation from the current 'gold standard' of manual gating.

Introducing the component of variability in manual analysis builds upon previous work that investigated operator variation in flow cytometry data analysis when following protocols and increasing gating strategy complexity [42, 43, 186]. However, this previous work was focussed on manual gating only and did not consider automated software data analysis.

Software comparison studies such as the FlowCAP challenges have not directly compared the variation from automated approaches to the variation from manual gating approaches, and instead relied on the F-measure as the population identification accuracy performance metric [49, 47]. Broader prior work on comparison of automated versus manual data analysis have been completed on a limited number of automated software tools [3], which this study aims to expand on.

At the start of this research, a participant study was planned to ask human operators to gate on previously generated synthetic and real-world data, to provide a direct comparison with the automated software analysis already performed. However, events of the Covid-19 pandemic and lockdown restrictions in the UK forced the plans for these

participant studies to be put on hold and revised. The scope of the research pivoted to sourcing existing participant analysed flow cytometry data, rather than running a new study and collecting novel data, with the requirement for these datasets already analysed by participants to be processed through software afresh. An alternative pool of participant data was obtained from UK National External Quality Assessment Scheme (NEQAS) for Leucocyte Immunophenotyping. This dataset is from a credible clinical source, generated following international standards [39, 40] and for the purposes of proficiency testing/ external quality assessments. This dataset from NEQAS had the benefit of being from a much greater number of participants, within a similar pool of laboratory setting, and tied in with the clinical laboratory survey carried out in Chapter 2.

In the study, participants were sent electronic files to analyse rather than biological test samples (i.e. stabilised peripheral blood) so that variation from reagents, sample processing, instruments could be excluded, and the focus was solely on variation arising from the data analysis/ gating process.

The structure of this chapter is broadly split into two parts with the NEQAS dataset at its core. First, is the automated analysis of the NEQAS dataset through different software platforms, leading to comparison of software performances and the validation of previously applied synthetic datasets as benchmarking toolsets. Second, is the comparison of automated analysis against participant laboratory analysis, with specific investigations into accuracy, variability, and correlation.

7.1.1 Chapter aims

The aims of this chapter are to:

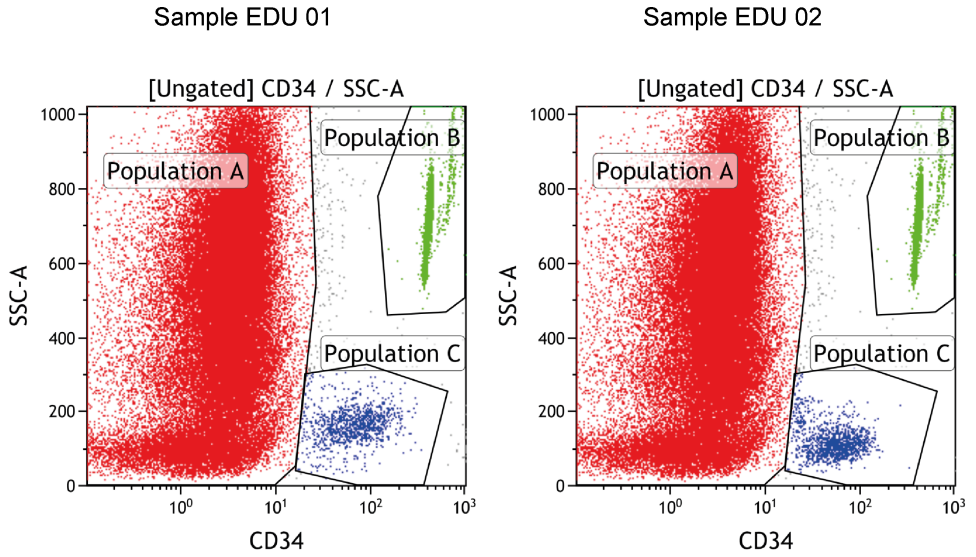
- Run different automated software through a real world flow cytometry data from a clinical setting.
- Use this clinical dataset to validate the software benchmarking approaches from previous chapters that used synthetic datasets.
- Compare the performance of cell population identification in automated software against laboratory participants.
- Investigate the correlation of cell counts of populations with different degrees of separation between software outputs versus laboratory participants.

7.2 Methods

7.2.1 NEQAS CD34 dataset

Data were obtained from the CD34+ stem cell enumeration electronic trials (EDU1–EDU13) issued by UK NEQAS-LI between May 2017 to October 2019. All samples were stabilised peripheral blood from patients who had undergone stem cell mobilisation prior to stem cell harvesting by apheresis. Informed consent was obtained from all patients by NHS Blood and Transplant prior to the donation procedure.

All electronic trials data were patient-derived with the exception of trial EDU2 which was artificially derived from trial EDU1 post-acquisition; the file was manipulated using FCS Express 6 (De Novo Software) to create different CD34+ counts (as Population C in Figure 7.1) while keeping the non-CD34+ and bead event counts the same (Figure 7.1, Populations A and B).



Gate	Number
All	100,000
Population A	86,918
Population B	11,367
Population C	1,058

Gate	Number
All	100,242
Population A	86,918
Population B	11,367
Population C	1,325

Figure 7.1: Electronic manipulation of sample EDU1 to generate sample EDU2. Counts for the CD34+ population (Population C) changed post-acquisition.

Samples were processed following International Society of Hematotherapy and Graft Engineering (ISHAGE) protocols [39, 40]. Briefly, samples were stained using antibodies CD45-FITC and CD34-PE (both BD Biosciences) in BD Trucount tubes containing fluorescent beads. Data were acquired on a BD FACSCanto II flow cytometer.

Additionally, samples for EDU6–EDU13 were also acquired on a Beckman Coulter Navios flow cytometer because of software incompatibility issues between BD and Beckman users that were identified during earlier trials. In these cases, Stem-Kit (Beckman Coulter) materials were used for sample processing. Within this research, although FCS files from two different cytometer manufacturers were available, only the ones from BD were taken forward for automated software analysis runs because of inconsistent parameters and naming conventions between the two instrument platform systems.

7.2.2 Data pre-processing

All pre-processing operations were performed in the R package *flowCore* v2.0.1 [134]. FCS files were filtered to discard margin events on the upper boundary of FSC and SSC channels. Scales in the fluorescent FITC and PE channels were linearised using the 'logicle' transformation function [216], with default parameters applied (linearisation width = 0.5, top of the scale = 262144, full width of transformed display = 4.5, additional negative range = 0).

7.2.3 Software runs

The input parameters used for software runs on the NEQAS CD34 dataset are listed in Table 7.1. For all software, channels used for clustering were: forward scatter (FSC), side scatter (SSC), FITC and PE. Software outputs were manually interpreted to select the target cluster as previously described. For SPADE3, outputs were partitioned into 8 to 11 sub-populations in a semi-automated manner using the 'auto suggest annotation' function, as previously described.

Software outputs were checked against a manual quality control (QC) criteria:

1. The identified CD34+ target population positioned in the correct region (all bright CD34+ events with low/mid SSC), based on ISHAGE industry standard gating (Figure 7.2).

2. The identified bead population visible with low FSC/ high SSC/ high fluorescence (in any channel).

3. The identified events formed of a single cluster that excluded non-target events.

Manual QC checks found that software tools were unable to correctly cluster the target population for a number of samples. In these cases, the runs were flagged as 'QC failed'

Table 7.1: User parameters for software runs on NEQAS CD34 dataset.

Software	**Parameter**	**NEQAS CD34 dataset**
Flock2	Bins	auto
	Density	auto
	Calculate centroids using	Mean fluorescence intensity
FlowMeans	Max number of clusters	10
FlowSOM	Number of expected metaclusters	10
	Grid size	10×10
PhenoGraph	*k*, initial clustering	15
	k, meta-clustering	5
SPADE1	Target number of nodes	10
	Downsampled events target	10%
SPADE3	Outlier density	1st percentile (default)
	Target density	20,000 cells (default)
	Number of desired clusters	100 (default)
SWIFT	Input cluster number	10
	Arcsinh transformation	0

and the best approximations for the target population count were reported based on operator judgement.

7.2.4 Separation index estimation

The separation index (SI) between the target cluster to the main cluster was estimated using the `sepIndex` function in the R package *clusterGeneration* [171]. Outputs from SWIFT and FlowSOM runs of the NEQAS dataset were used to estimate the CD34+ and bead cluster SIs, respectively.

7.2.5 Electronic trials participant data

Electronic trials were distributed to laboratories participating in the UK NEQAS-LI CD34 stem cell enumeration programme, as an optional educational exercise. Participants were asked to analyse the files as per their in-house procedures and to submit results for the total CD45+, CD34+, and bead event counts, the percentage CD34+ cell count, and the absolute CD34+ cell count (cells/μL). All participating laboratories stated they followed the ISHAGE gating strategy [39, 40]. Outliers from participant reported data that fell one order of magnitude away from the mean were excluded from further analysis — these

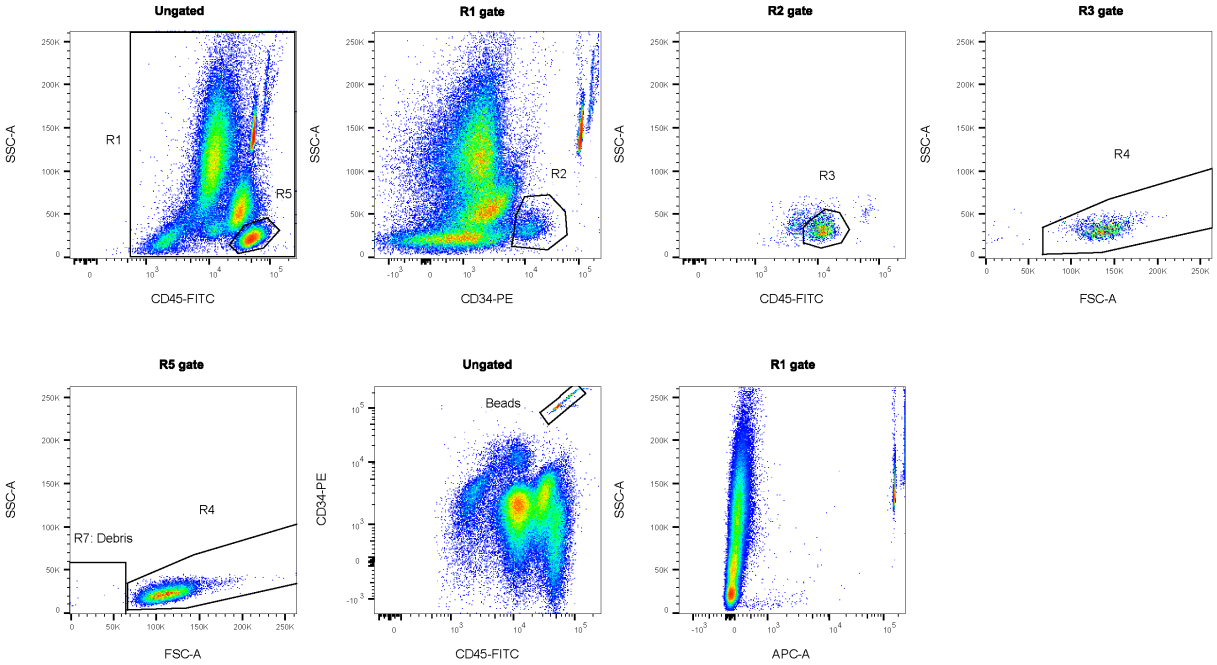


Figure 7.2: Example of ISHAGE gating on NEQAS dataset of stabilised peripheral blood. CD34+ stem cells are enumerated by gating on all CD45+ events (R1), followed by all CD34+ events with low/mid SSC (R2). R3 is used to adjust the cell population to delineate true CD34+ cells. The cells are further adjusted in the lymph-blast R4 gate, the position of which is set according to the lymphocytes gated in R5 in the R1 region. Beads can be gated using the last two plots shown here. Viability testing is not used for analysis of stabilised cells.

results exceeded the total number of events in the sample and were clearly incorrect.

7.2.6 Statistics

Methods used for statistical analysis included the mean, standard deviation (SD), and coefficient of variation (CV). All statistical tests were performed in R version 4.0.4.

For performance assessment, the absolute difference between software outputs (CD34+ cell and bead populations counts) to a reference value was calculated using Eq. 7.1,

$$\text{Difference to reference } = |A - B| \tag{7.1}$$

where A is the software count, and B is the mean of laboratory participant reported counts.

In keeping with international standards for proficiency testing ISO 13528:2015 [217], the z-scores for software outputs were calculated to enable comparisons of the deviation from the mean, as in Eq. 7.2:

$$z = \frac{x - \mu}{\sigma} \tag{7.2}$$

where x is the software count, μ is the mean of the participant counts, and σ is the standard deviation of the participant counts.

Comparison between the means of participants and software reported counts for each of the 13 samples was performed using the unpaired t test, assuming counts follow a Gaussian distribution and have equal variances.

The Mann-Whitney test (also called the Wilcoxon rank sum test) was used to compare the distributions and medians of the CV values between the two participants and software groups, assuming that the CV data are non-parametric.

Correlation between software and participant analyses was measured using Pearson's correlation coefficient, r [218], with the assumption that the mean target population counts from the 13 samples follow a Gaussian distribution.

7.3 Results

7.3.1 Results of NEQAS dataset software runs

The NEQAS dataset was processed through seven different automated flow cytometry data analysis software (Flock2, flowMeans, FlowSOM, PhenoGraph, SPADE1, SPADE3 and SWIFT), and manually checked against the QC criteria defined in Section 7.2.3. Four channels were used for clustering the data (FSC, SSC, CD45-FITC and CD34-PE), in contrast to previous work with synthetic datasets, where only two channels were available. Furthermore, input parameters were deliberately chosen to cause the software to return approximately 8 to 10 clusters (rather than 2 or 3 for synthetic datasets).

7.3.1.1 Clustering characteristics

As illustrated in previous chapters, different clustering characteristics were observed from the different software. Here, visualisation of the clustered data in multiple 2D plots with different parameter combinations offered additional insights into the varied partitioning approaches implemented by the different software.

'Soft' partitions of the data were observed for Flock2, FlowSOM and SWIFT (Figure 7.3B, D, H), with clusters having indistinct boundaries, and slightly overlapping each other in all combinations of the 2D plots. Certain horizontal and vertical 'cut-offs' were

apparent in SWIFT outputs, giving indications of the process it applied to discard data (Figure 7.3H).

'Hard' partitions in the data were more apparent for flowMeans, PhenoGraph, SPADE1 and SPADE3 when viewed in the FSC vs. SSC plots (Figure 7.3C, E, F, G). Straight line divisions between clusters were observed for flowMeans, while characteristic meandering boundaries between clusters were observed for PhenoGraph, SPADE1 and SPADE3. The hard boundaries were less visible in the CD34-PE vs. SSC plots, which could suggest that the partitioning applied by these software were more weighted towards the two scatter parameters rather than the fluorescent ones.

7.3.1.2 QC passes and fails

Based on the manual QC criteria described in Section 7.2.3, SWIFT was the best performing software tested here for identifying the CD34+ target population. In 11 out of 13 samples (85%), SWIFT was able to successfully identify the CD34+ population in the correct region on plots as a single cluster (Figure 7.4A). A closer inspection of the failed SWIFT runs from samples 7 and 12 revealed these CD34+ populations had lower event counts than other samples, and were also challenging to gate manually, as demonstrated by the high CVs from participants data (Table 7.4). FlowSOM and Flock2 had lower CD34+ QC pass rates of 7 in 13 (54%) and 6 in 13 (46%), respectively. Unsuccessful runs were caused by identification of non-target events by the software which on visual inspection were actually separate from the core CD34+ cluster (as an example, see Figure 7.3D). Four of the software tested in this study (flowMeans, PhenoGraph, SPADE1 and SPADE3) were unable to identify the target CD34+ population in all 13 samples of the NEQAS dataset. In most cases, the CD34+ events were grouped with other events with similar FSC/SSC features, and appeared in the CD34-PE vs. SSC plots as a horizontal smear (Figure 7.3C, E, F, G).

For the bead population, five of the software tested (Flock2, flowMeans, FlowSOM, SPADE1 and SPADE3) were able to successfully identify the cluster in all 13 samples (Figure 7.4B). Contrastingly, SWIFT failed to isolate the bead population throughout all 13 samples. The highly elongated shape of the cluster may have potentially posed problems for the SWIFT algorithm that identifies clusters based on a Gaussian mixture model fitting strategy [88]. Identification of the bead population by PhenoGraph was not always straightforward, noting that one of its runs failed (sample 5).

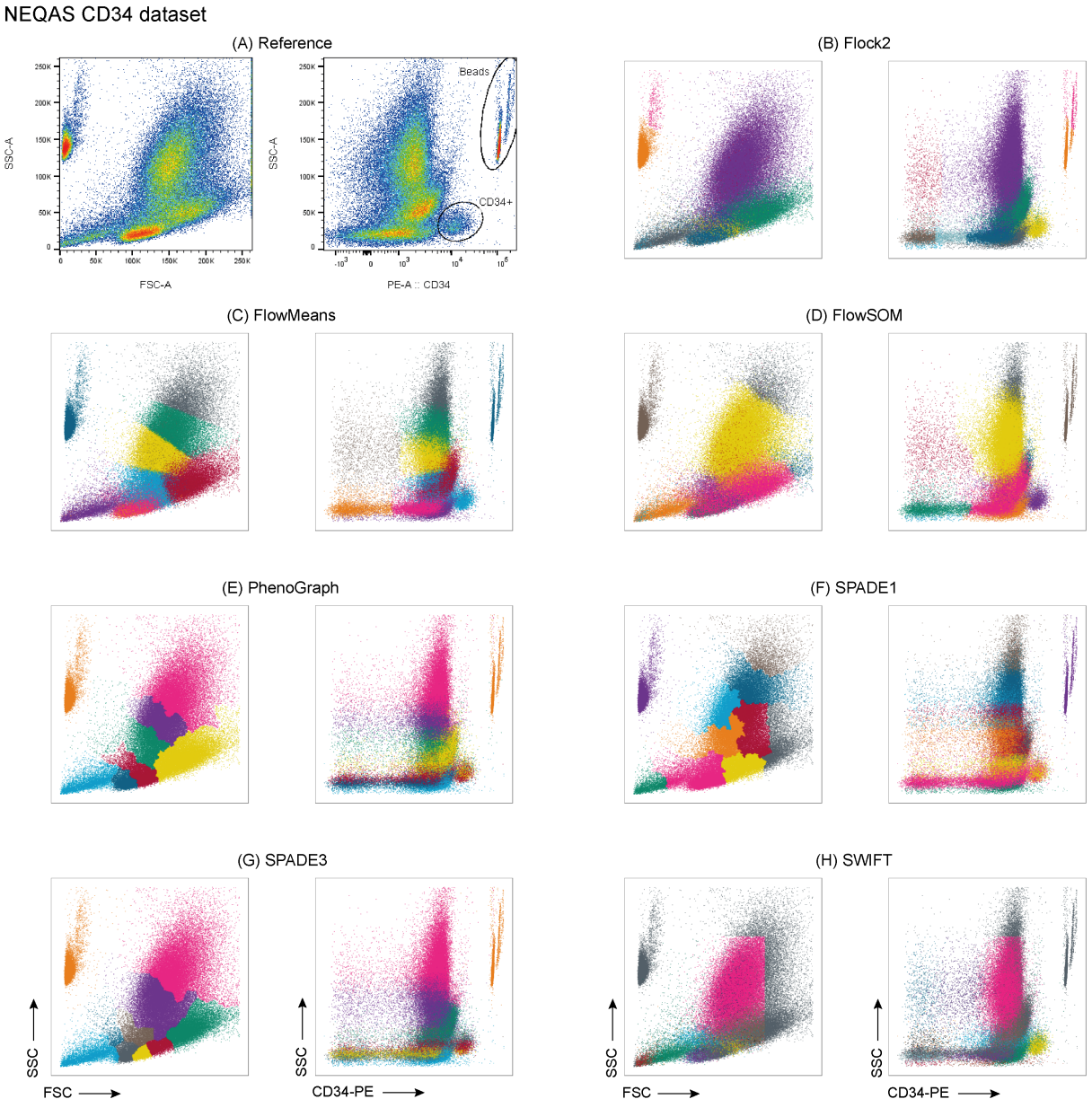


Figure 7.3: Representative clustering outputs from runs of the NEQAS CD34 dataset on different software.

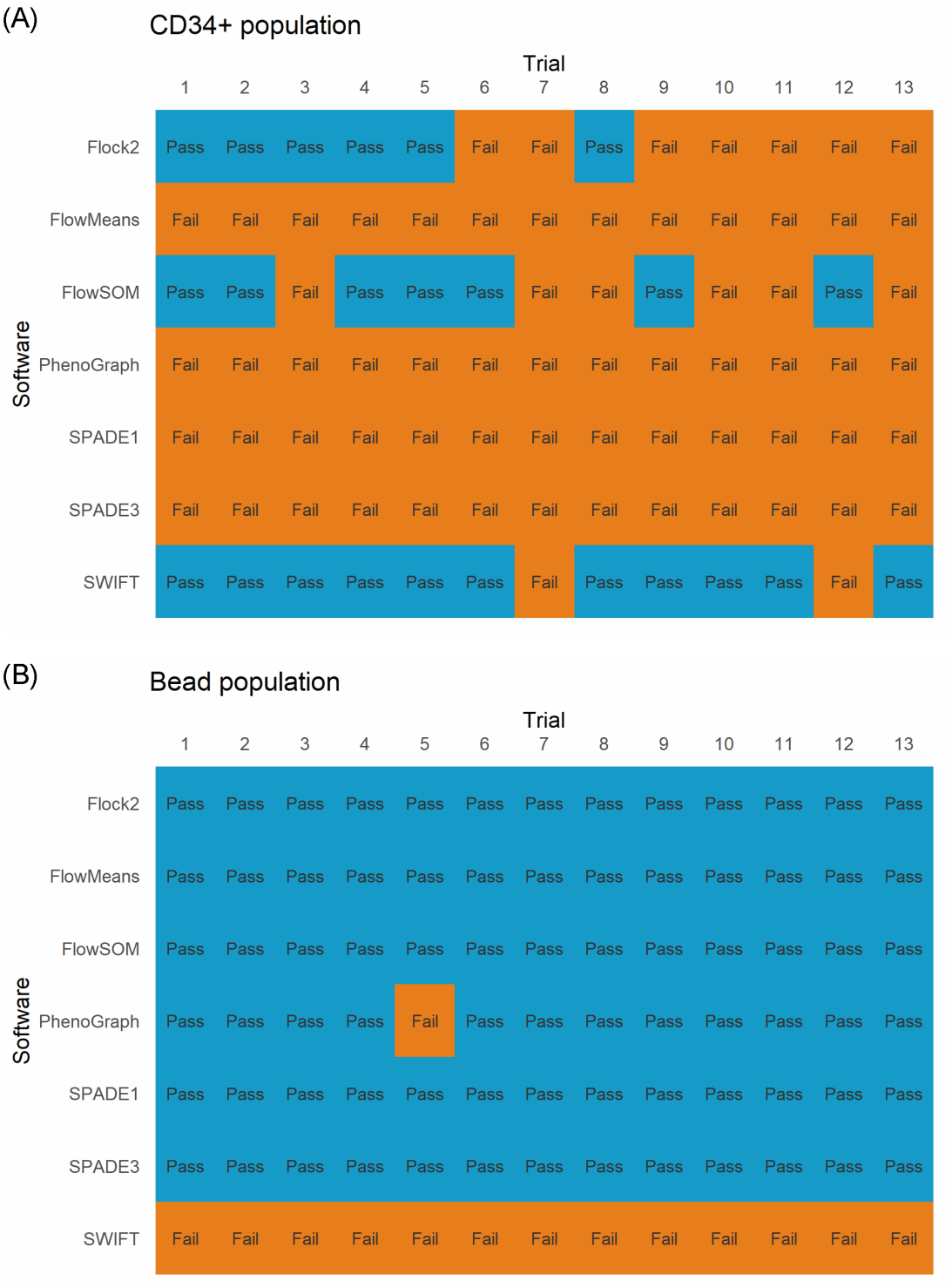


Figure 7.4: Heatmaps of manual QC results.

7.3.1.3 Performance analysis

The software output difference to reference value was calculated to assess the automated cell population identification performances. All outputs were used for this analysis, including ones that failed manual QC checks. The means of the laboratory participants event counts were used as the reference value.

The performances of software varied between each other for the identification of the CD34+ population. SWIFT results displayed good accuracy and repeatability for the CD34+ population, reporting a population percentage difference to reference of $(0.17 \pm 0.19)\%$. In contrast, SPADE1, PhenoGraph and SPADE3 reported the largest difference to reference values of $(12.2 \pm 6.2)\%$, $(11.5 \pm 4.0)\%$, and $(8.8 \pm 6.9)\%$, respectively. Flock2 and FlowSOM analyses produced noticeably large error bars for the difference to reference of $(8.3 \pm 15.5)\%$ and $(6.3 \pm 12.5)\%$, respectively. The reduction in software repeatability observed here was most likely caused by those outputs that failed QC checks.

For the bead population, results from this analysis showed that all software outputs had high levels of accuracy and repeatability, with differences to reference reported from all software falling below $(0.16 \pm 0.13)\%$ excluding SWIFT, which failed to identify the bead population so its performance for that task could not be assessed (Figure 7.5).

Overall, the software outputs for the CD34+ cells were less accurate and repeatability worsened compared with the bead population.

Further insight into the variation of software outputs was provided by calculation of a z-score, or the number of SDs the output lies above or below the mean (assigned here as the value from participant counts). Results of the z-scores are summarised in and Figure 7.6 and Table 7.2, show clear differences in the high variation of CD34+ cell counts, the majority of which were over 3 SDs away the mean, contrasted with the low variation of bead counts (all within 2 SD of the mean).

Based on conventional proficiency testing interpretation of the z-scores [217], all the bead counts from all software were calculated to have $z \leq 2.0$ and would be considered 'acceptable' results, with the exception of SWIFT which returned no outputs, and one PhenoGraph run (Table 7.3).

For the CD34+ cell counts, 11 SWIFT outputs, along with six outputs from each of FlowSOM and Flock2, and one result from SPADE3 gave $z \leq 2.0$ and would be considered 'acceptable'. One output from Flock2, SPADE3, and SWIFT apiece had z-scores between 2.0 and 3.0 and would be given 'warning signals'. However, the majority of software outputs for the CD34+ cell counts, including all those from flowMeans, PhenoGraph and SPADE1, gave $z \geq 3.0$ (medians ranging from 51 to 111, Table 7.2) and would be

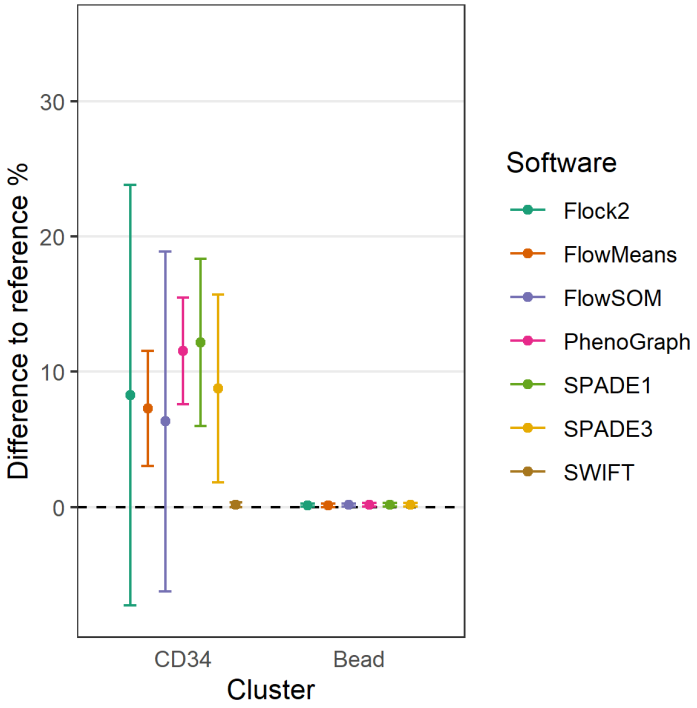


Figure 7.5: Summary of software performance for detection of target CD34+ cell and bead populations. N=13, mean ±1SD, dashed line = target.

considered 'unacceptable' (Table 7.3). The results from the z-scores largely agreed with the manual QC interpretations of software outputs (Figure 7.4).

In terms of performance rankings, on the basis of the z-scores across all samples, SWIFT was ranked as the best performing for identification of CD34+ cells, followed by FlowSOM then Flock2. flowMeans and SPADE3 were tied in fourth place, followed by PhenoGraph, and finally SPADE1 was ranked last. Meanwhile, for the bead population, flowMeans was ranked in first place followed closely by Flock2, FlowSOM, SPADE1, SPADE3, and PhenoGraph. SWIFT was ranked in last place here for failing to identify the bead populations.

7.3.1.4 Separation index estimation

There appeared to be a clear difference in software performance between the identification of the well-separated bead cluster and the CD34+ cluster, which was closer to other cell populations. To place the different degrees of separation of the two clusters in the context of previous work using a synthetic separation dataset (Chapter 4), the separation index (SI) was estimated for both the target clusters in the NEQAS dataset (Figure 7.7).

A problem with calculating the SI from real data is that the clusters have to be

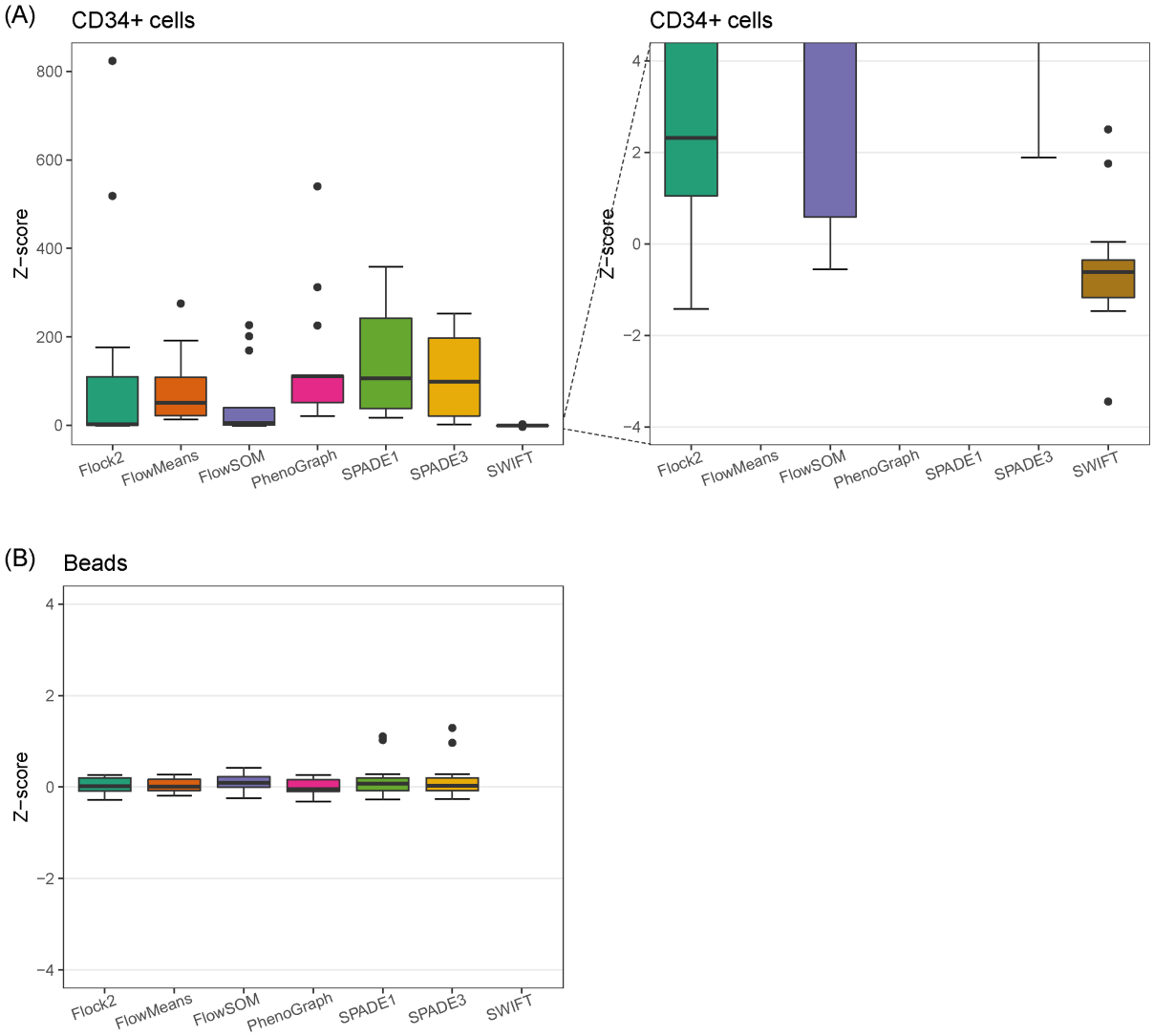


Figure 7.6: Box-plots of z-scores for (A) CD34+ cells and (B) bead populations. Median; bounds, 25th and 75th percentile; whiskers, smallest and largest value within 1.5 times the inter-quartile range; individual points, outliers.

Table 7.2: Median z-scores

Software	**N**	**CD34+ cells**	**Beads**
Flock2	13	2.31	0.02
flowMeans	13	51.41	0.01
FlowSOM	13	5.56	0.09
PhenoGraph	13	110.81	−0.05
SPADE1	13	106.07	0.08
SPADE3	13	98.92	0.03
SWIFT	13	−0.62	NA

Table 7.3: Interpretation of software outputs based on z-scores.

		CD34+ cells			Beads	
Software	**N**	**Acceptable**[1]	**Warning**[2]	**Unacceptable**[3]	**Acceptable**[1]	**No output**
Flock2	13	6	1	6	13	-
flowMeans	13	-	-	13	13	-
FlowSOM	13	6	-	7	13	-
PhenoGraph	13	-	-	13	12	1
SPADE1	13	-	-	13	13	-
SPADE3	13	1	1	11	13	-
SWIFT	13	11	1	1	-	13

[1] $|z| \leq 2.0$

[2] $2.0 < |z| < 3.0$

[3] $|z| \geq 3.0$

partitioned first, and because different methods produce different outputs, there is a decision to be made for which ones to use to estimate the SI. Since most software (except SWIFT) demonstrated good performance when clustering the bead population, FlowSOM outputs were arbitrarily selected to estimate the bead cluster SI values.

The bead clusters had a mean SI of +0.26, indicating good separation, with a range between −0.03 and +0.64 indicating just touching to very well-separated clusters. This range arose because although the bead cluster was well-separated in certain parameters (e.g. CD34-PE channel), it was less well-separated in the CD45-FITC marker channel (Figure 7.8). An important difference affecting SI estimation is highlighted here, because whilst only two parameters were previously used for the synthetic data analyses, four parameters were used here for real-world data analyses.

For the CD34+ clusters, outputs from SWIFT were used for SI estimation, because SWIFT demonstrated the best performance for identifying this population. The CD34+ clusters were generally less well-separated, with a mean SI of +0.07, indicating clusters touching, and a range between −0.09 and +0.21 indicating overlapping to well-separated clusters.

7.3.1.5 Link to synthetic datasets

The pattern of performance observed here, where automated software identified cell populations with better accuracy and repeatability as clusters became more well-separated,

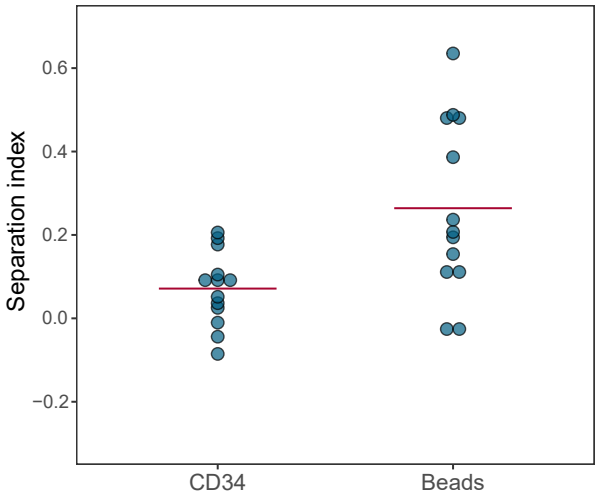


Figure 7.7: Separation index estimation of CD34+ cell and bead clusters. Red bar, mean; circles, individual data points.

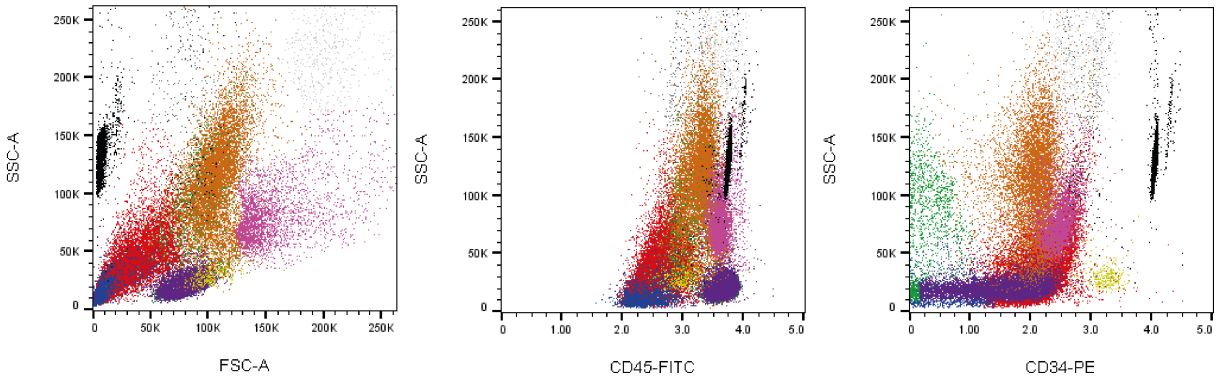


Figure 7.8: The bead cluster (in black) displayed different degrees of separation in different marker channels.

reflected previous results shown with synthetic datasets in Chapter 4 (Figure 7.9).

Software performance of the real-world bead cluster having a SI of +0.26 matched those of clusters with SIs between +0.2 to +0.3 from the synthetic two-cluster separation dataset.

Interestingly, the difference to reference results at the estimated SI for the CD34+ population (Figure 7.5) were not directly comparable to results in the synthetic dataset between a SI of 0 and +0.1, but were more alike to results at a SI of −0.2. This result suggests that the software performance was worse with analogous real-world data of similar SIs but of higher parameter/data complexity, or raises the question of whether the SI estimation, based here solely on best-performing SWIFT outputs, could be optimised to better represent the real gap between clusters.

Nevertheless, taken together, the trends in deterioration of software performance as clusters come closer together have been clearly illustrated in this work with both real-world and synthetic datasets.

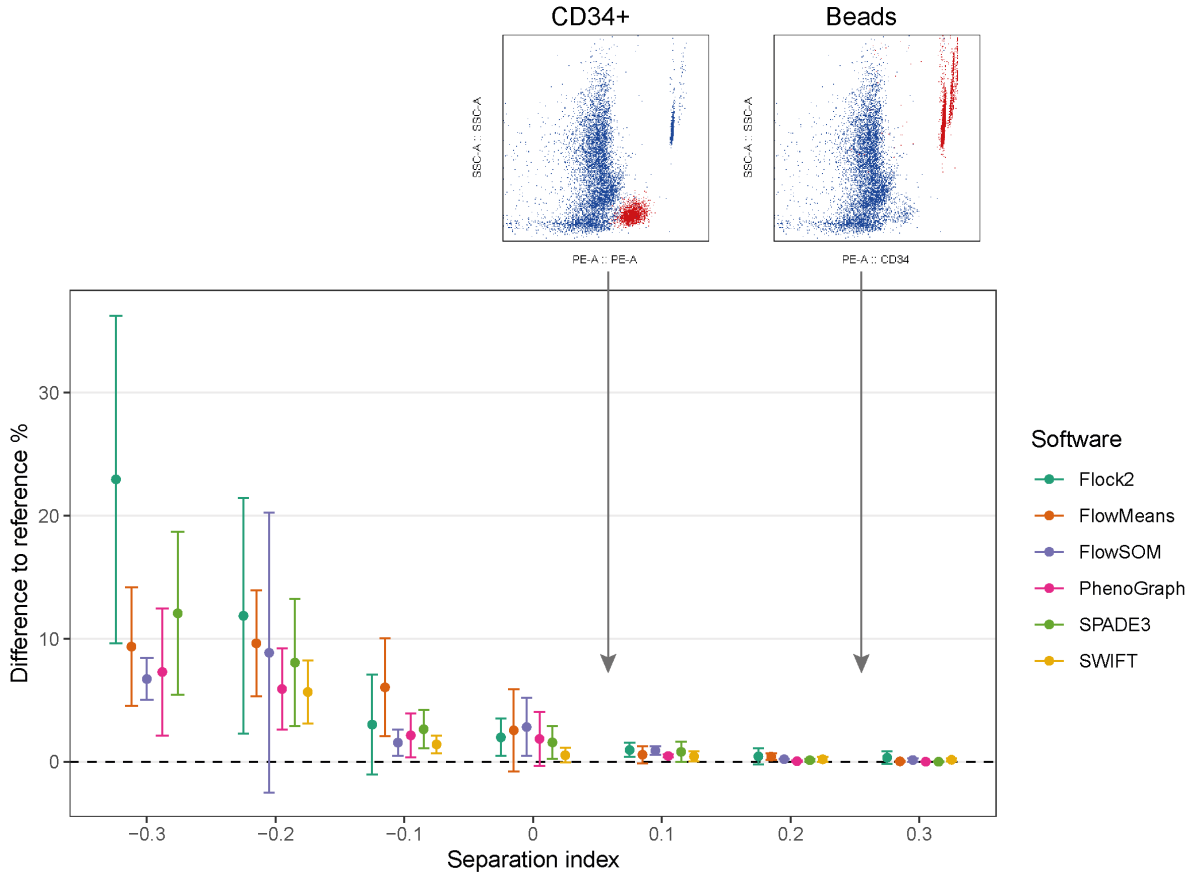


Figure 7.9: Degree of separation of the real-world CD34+ cell and bead populations, overlaid in relation to software performance results from analysis of synthetic two-cluster separation dataset from Chapter 4.

7.3.2 Results of electronic trial participant data analysis

Statistics for participants' analysis of the 13 NEQAS electronic trials are summarised in Table 7.4 for the CD34+ cell counts, and Table 7.5 for the bead counts. Only results from BD users are included for samples 6 to 13.

Note that files from samples 1 and 2 originate from the same donor material, but sample 2 was generated from modification to the FCS file from sample 1, to create different CD34+ counts while preserving the same non-CD34+ and bead event counts (Figure 7.1).

7.3.3 Comparison of software vs. participant data analysis

7.3.3.1 Total event counts

The total event counts from all software outputs were grouped together and compared with the counts reported from all the participants, for each sample in the study.

The CD34+ cell counts from automated software were higher than those from participants, and showed greater variability across all 13 samples (Figure 7.10A). For example, the mean CD34+ count returned from sample 9 was $16,404 \pm 17,984$ from software compared to 798 ± 63 from participants, a difference between means of 15,606 events. For comparison between the two analytical methods, calculation of unpaired t tests from each sample revealed highly significant differences between the mean CD34+ counts of all the runs ($p \leq 0.0001$).

When considering only software outputs of CD34+ counts that passed manual QC checks, the differences between software and participant reported counts were reduced (Figure 7.10B). Taking sample 5 as an example, where three software passed the manual QC checks (Flock2, FlowSOM and SWIFT), the mean CD34+ count from software was $1,768 \pm 70$ compared to $1,733 \pm 71$ from participants, a difference of just 35 events. The majority of samples had no significant differences between participants and software mean CD34+ counts, with the exception of samples 2 and 9 (both $p < 0.05$).

There were fewer differences in the mean bead counts between participant and software outputs across all 13 samples, and software outputs had lower variability (Figure 7.10C). Using the same example of sample 9, the mean bead count was $4,999 \pm 11$ from software compared to $5,011 \pm 97$ from participants, a difference of just 12 events. Analysis of the difference between the means for each sample showed all of them had no significant differences between the participants and software outputs ($p > 0.05$).

These results show that while manual analysis methods outperformed automated analysis for identification of the CD34+ cells (in terms of repeatability), the reverse was

Table 7.4: Statistics for participants' total CD34+ event counts.

Sample	N	Min	Max	Mean	StDev	CV
1	102	135	1,203	893	107	12
2	102	345	1,286	1,082	118	11
3	84	19	1,454	1,186	212	18
4	83	332	988	385	76	20
5	83	1,479	1,867	1,733	71	4
6	55	76	517	270	54	20
7	55	3	972	70	160	230
8	55	566	1,220	857	95	11
9	55	576	903	798	63	8
10	45	202	1,212	919	153	17
11	45	2	2,411	1,874	421	22
12	45	3	1,887	122	382	314
13	45	117	261	169	26	15

Table 7.5: Statistics for participants' total bead event counts.

Sample	N	Min	Max	Mean	StDev	CV
1	96	2,000	47,900	11,521	3,873	34
2	95	1,451	47,900	11,497	4,120	36
3	83	1,020	9,884	9,401	1,249	13
4	82	1,020	5,049	4,845	443	9
5	82	1,020	2,851	2,660	194	7
6	54	1,078	7,065	3,974	595	15
7	55	13,226	17,821	17,512	642	4
8	55	6,701	7,243	7,033	123	2
9	54	4,773	5,171	5,011	97	2
10	45	8,405	17,928	17,024	1,424	8
11	45	7,496	10,346	10,078	468	5
12	45	7,047	16,151	15,643	1,370	9
13	45	6,726	10,804	7,405	561	8

apparent for identification of the beads.

7.3.3.2 Coefficient of variation

Following analysis of the total event counts above, the CV was calculated to allow direct comparisons on the distribution of outputs between the two analytical approaches for the two target population groups.

For the CD34+ cell counts, the CVs from participants ranged from 4% to 314%, with a median of 17% (Figure 7.11A). The wide inter-laboratory CV range reported here are comparable to those from previous external quality assessment studies [219]. The CVs from participants were noticeably high for samples 7 and 12 (230% and 314%, respectively), and was possibly a function of the lower cell numbers in these samples. The CVs from software ranged from 65% to 138%, with a median of 98%. Comparison of the CVs between the two groups show that, in this instance, software CVs were significantly higher than those from participants ($p < 0.01$; Mann-Whitney test). This result indicates that the identification of the target cell population by software was not able to match the repeatability from manual gating.

For completeness in the analysis of variation in the CD34+ cell counts, the CVs from only software that passed QC checks was calculated, and was found to range from 4% to 65%, with a median of 15% (Figure 7.11B), noting that this statistic could not be calculated for 5 of the samples because fewer than 2 runs gave acceptable outputs. The software CVs from this analysis were more similar to those from the participants. The reduction in variability once failed runs were excluded was to be expected, and suggest that manual intervention or review of software outputs may be a viable solution for integration of automated tools into data analysis workflows.

For the bead event counts, the CVs from participants ranged from 2% to 36%, with a median of 8.4%. In comparison, the CVs from software ranged from 0.1% to 3%, with a median of 0.3% (Figure 7.11C). The difference in CVs between the two groups was found to be significant ($p < 0.01$; Mann-Whitney test). The lower CVs from the software outputs indicate more repeatable performance over manual methods when identifying distinct well-separated populations, and perhaps a lesser need for manual checks. Both participants and software groups gave improved CVs for analysis of bead counts in comparison to that of the CD34+ cell population, affirming that this population was more straightforward to gate.

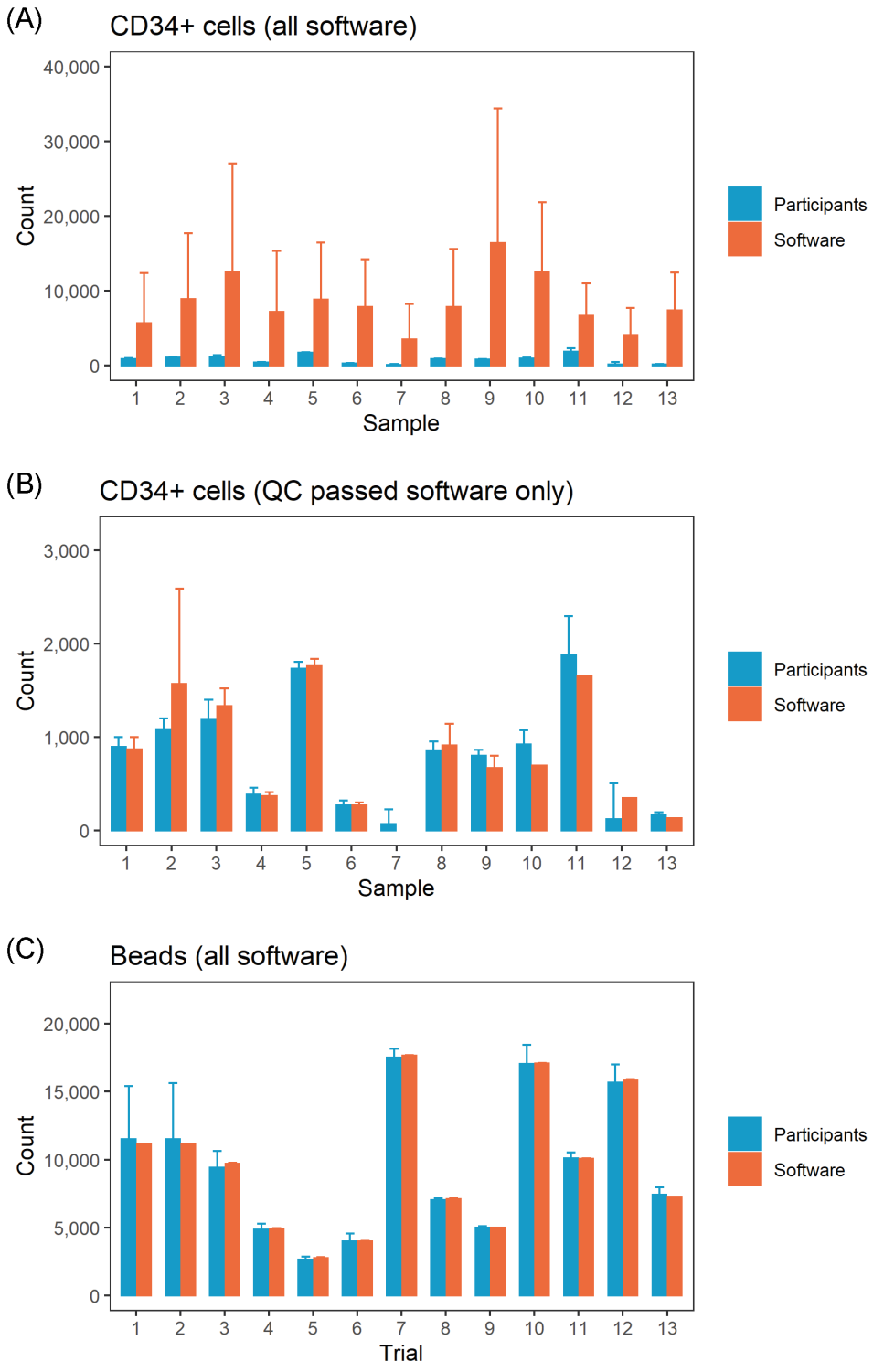


Figure 7.10: Participant vs. software comparison of target population counts. Bead counts included from all software with the exception of SWIFT and one PhenoGraph run which did not return any bead values.

7.3.3.3 Correlation

Finally, to understand the strength of the relationship between participants and software outputs, Pearson's correlation coefficient (r) was calculated using the mean event count from each of the 13 samples (assuming Gaussian distribution of the values), for both CD34+ cell and bead populations.

There was no correlation seen for mean CD34+ cell counts between participants and software outputs ($r = 0.332$, $p = 0.267$), which was expected given that outputs from software were significantly higher than those of participants (Figure 7.12A). However, when excluding software runs that failed QC checks, a strong association for the CD34+ cell counts between the two analytical approaches emerged ($r = 0.941$, $p < 0.0001$) (Figure 7.12B). In contrast, there was a significant and near perfect correlation for bead counts between participants and software ($r = 0.999$, $p < 0.0001$) (Figure 7.12C), indicating a strong agreement between the two methods.

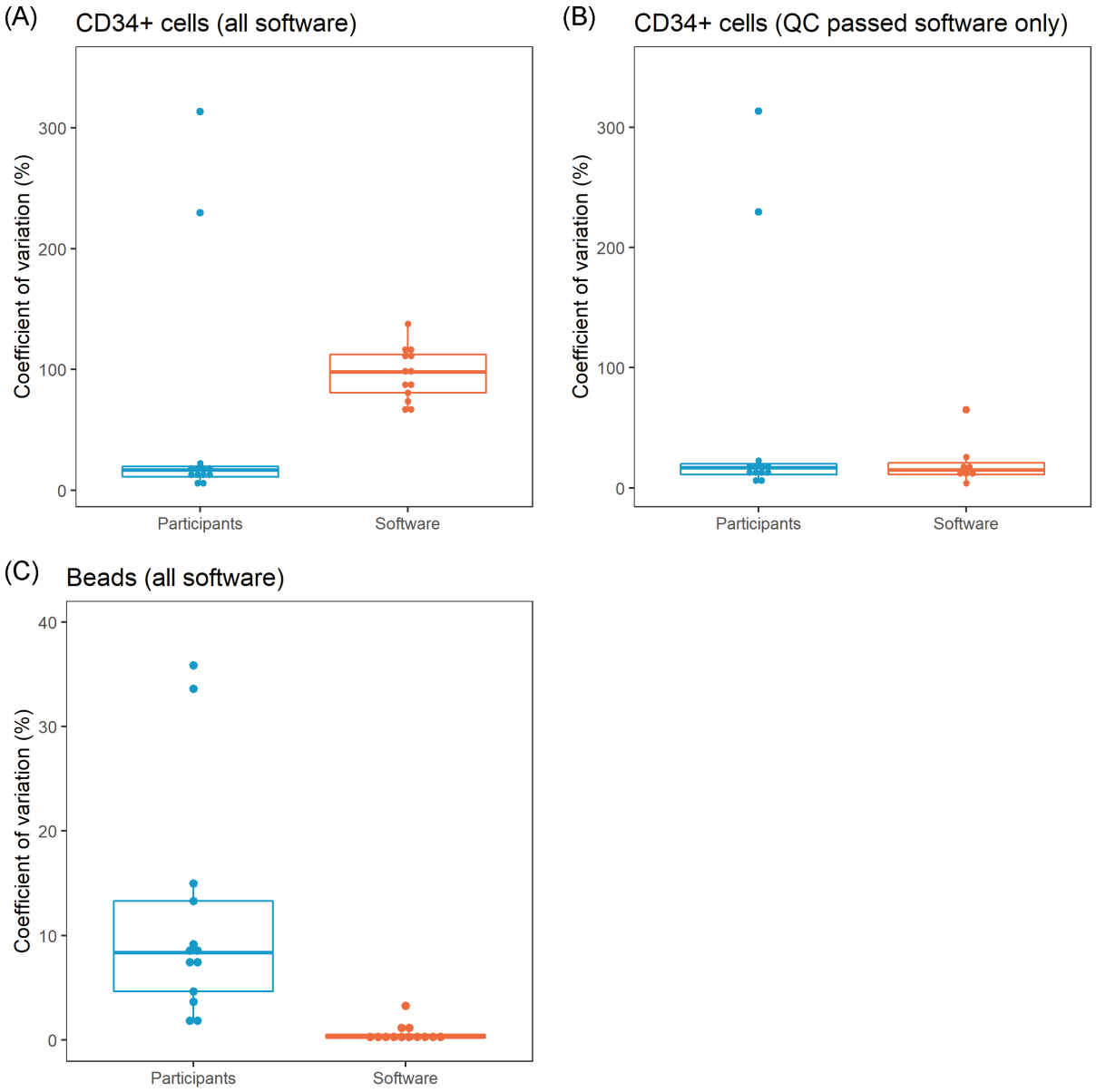


Figure 7.11: Participant vs. software comparison of target population CV.

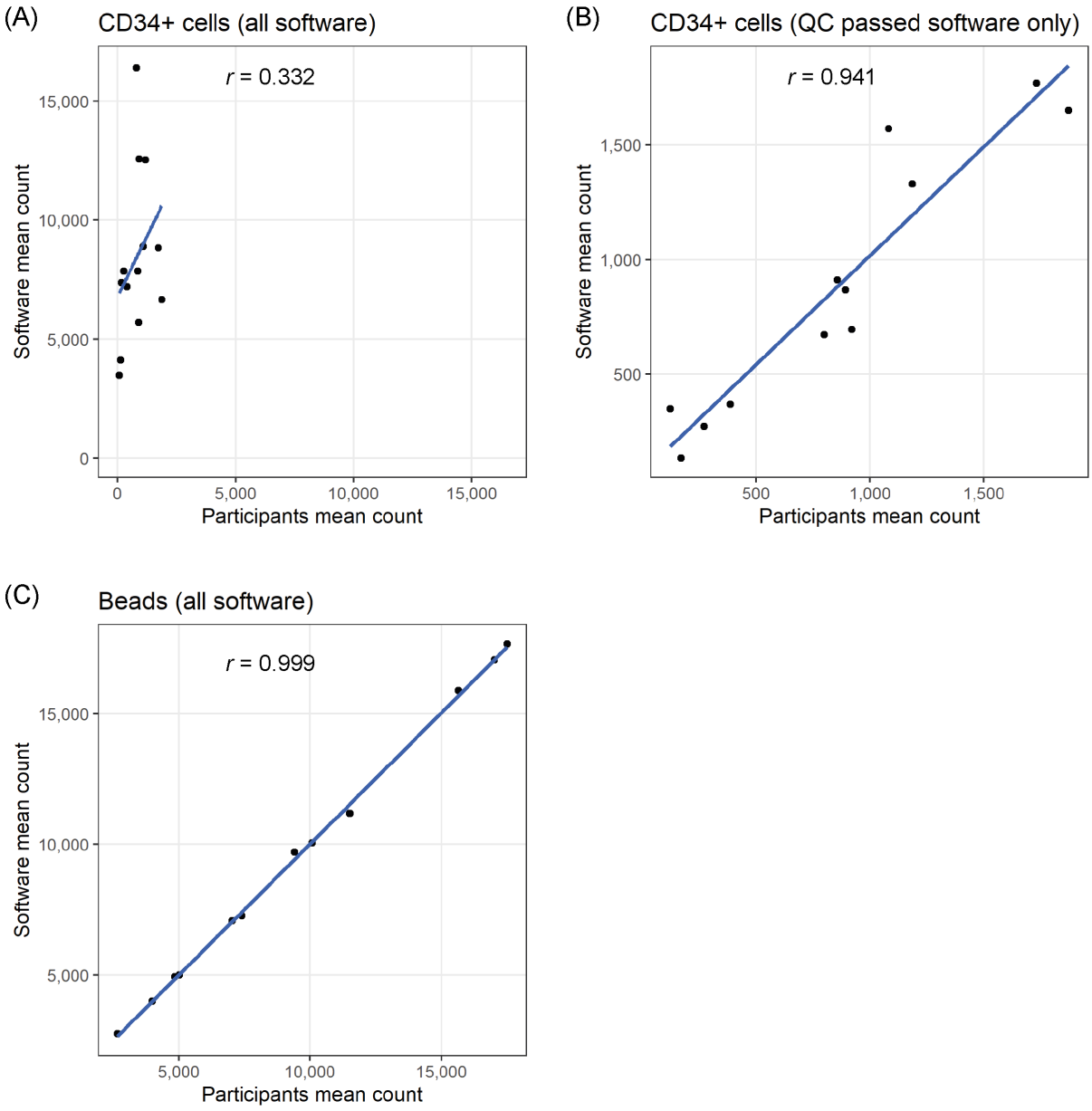


Figure 7.12: Correlation between manual and automated software analysis approaches for mean target population counts. Bead counts included from all software with the exception of SWIFT and one PhenoGraph run which did not return any bead values.

7.4 Discussion

Within this Chapter, two key work components have been completed through application of a real-world flow cytometry dataset from the UK NEQAS-LI programme for CD34+ stem cell enumeration.

Firstly, the dataset was processed through seven different automated software to detect two target populations: the CD34+ cells and the beads. These two populations had different degrees of separation, with the beads clusters being more distinct and well-separated, whereas the CD34+ cell clusters were closer to or slightly touching the bulk non-target populations. The aims here were to assess software performance in accuracy and repeatability, and to validate the benchmarking approach using synthetic datasets from previous chapters.

The investigations found similar patterns in the deterioration of software performance as clusters moved closer together, and whilst the beads were easily identified by all software (except SWIFT), much of the automated CD34+ cell count outputs from software such as PhenoGraph, SPADE1 and SPADE3 failed the manual QC criteria, and when evaluated using the z-score, were classed as 'unacceptable'. SWIFT demonstrated the best performance when identifying the CD34+ cell population. The results can be directly compared to benchmarking outputs using synthetic datasets, and give further evidence for the use of well-designed and controlled synthetic datasets for quality assessments in computational flow cytometry.

The second component of this Chapter has addressed the comparison of variation between automated and manual data analysis methods. Total event counts from 45 to 102 laboratory participants were compared to counts from the seven software that apply different unsupervised clustering algorithms. Differences were observed between quantification of the CD34+ cells and that of the well-separated beads cluster. For the CD34+ cells, the variability from participants were significantly lower than software outputs, and there was weak correlation between the two analysis methods. In contrast, for the bead population, the variability from software analyses were significantly lower compared to manual methods, with a very strong correlation between the two measurement approaches. These results demonstrated that while well-separated clusters can be analysed with a high level of confidence using automated tools, the software identification of slightly touching cell populations may not yet match the potential quality of manual methods.

Although the findings here are consistent with previous work that found higher variability associated with poorly-resolved cell populations compared with clearly resolved ones [178, 201, 3], this is the first study of its kind to compare large scale inter-laboratory

variability with inter-software variability.

Furthermore, this study benefitted from using only electronic data files, so that variability from gating/data analysis was specifically isolated from other upstream sources of variation in flow cytometry e.g. antibody reagents, instrument variation, and sample processing.

It was outside the scope of this study to explore any reduction in manual workload or increased efficiency in processing samples when incorporating automated data analysis software. However, during this work and as observed in other Chapters, significant efforts were required to cause automated tools to output cell counts, and also to manually QC check all software outputs. Thus, adoption and implementation of automated tools would likely see a shifting of workload from manual gating to reviewal, adjustment and approval tasks.

A recognised limitation of this study is its focus on unsupervised clustering algorithms and the exclusion of supervised learning algorithms such as FlowDensity [87] or FlowLearn [102], and Infinicyt software [148] (which the previous survey of clinical laboratories found was already in use for automated analysis of haematological malignancies). The major challenge with including supervised learning tools in studies such as these would be the massive datasets needed for training and validation, and the question of how to generate such synthetic datasets with multi-factorial components. This would be a significant undertaking outside the scope of the current work, but would form a potential area for further research.

The current work on comparability between manual and automated data analysis presented here has opened up several potential avenues of further work. The utility of synthetic files as surrogates for real flow cytometry datasets can be further demonstrated, both in benchmarking studies and as a resource in an educational setting.

Distribution of the synthetic datasets with controlled properties (separation between clusters, rare populations, etc.) generated within this Thesis to participants to manually analyse would provide a better understanding of how human operators process synthetic data, and allow for a direct comparison to software variability observed in previous chapters. There is also scope for having operators analyse data in a controlled environment following standard protocols, rather than having the freedom of using 'local procedures'. Along these lines, a further possibility is for operators to process the data using the same automated workflow, so inter-operator variation of software implementation can be explored.

Investigation of different real world cell populations, such as the variation between automated and manual methods when analysing rare populations, would be of particular

interest given their importance in clinical and manufacturing settings. Files from NEQAS programmes on minimal residual disease or paroxysmal nocturnal haemoglobinuria (PHN) could be potentially used for this work.

In summary, automated tools have the potential to reduce operator variability, but with multiple software solutions available, this study has shown the extent of inter-software variation when analysing cell populations that are close together is significantly higher than the inter-laboratory variability. Careful consideration should therefore be given to implementation of automated cell population identification systems in high-throughput clinical settings, and the opportunity to increase the efficiency of analytical workflows through automation should be balanced with maintaining the high quality of current manual gating methods.

7.5 Chapter conclusions

- A CD34+ stem cell dataset issued by UK NEQAS-LI was processed through different automated software.
- The results validated the patterns in software performance trends seen from previous synthetic dataset runs, where well-separated clusters (beads) were identified with better accuracy and repeatability compared to closely touching clusters (CD34+ cells).
- Comparison between automated and manual data analysis methods showed a difference between quantification of the CD34+ cells and that of the well-separated beads cluster. For the CD34+ cells, the median CV from software was 98% compared to 17% from participants. In contrast, lower variability was shown for software analysis of bead counts compared to manual methods (median CVs of 0.3% and 8.4%, respectively).
- The study found a weak correlation between the two analysis methods when identifying the CD34+ cells ($r = 0.332$), but a very strong correlation between them for the beads ($r = 0.999$).
- Significant manual quality control checks were required to flag software outputs that failed to correctly cluster the target population.

Chapter 8

Conclusions and further work

8.1 Summary of the thesis

Automated flow cytometry data analysis software tools have the potential to improve the quality of cell therapy product characterisation through the reduction in process variation arising from manual gating. However, their widespread adoption in the biomanufacturing community is potentially restricted by the lack of clarity on the internal workings of the software algorithms, combined with the lack of tools that can be used to assess the confidence in software derived measurement outputs. Consequently, the aim of this thesis was to define the confidence in flow cytometry automated data analysis software tools. It is worth stressing that this was a comparison of techniques rather than a promotion of any one technique towards its optimal performance.

To better understand the current landscape of computational tools available for flow cytometry users, a literature survey was conducted at the beginning of this research, in Chapter 2. The results of the survey identified the most cited and most used software among flow cytometry users, and enabled the selection of software tools representative of the different algorithmic techniques available, and relevant within academic and clinical settings, for the comparison studies performed in the rest of this thesis. This initial selection was checked and updated throughout the thesis period, without addition of new tools on the basis of similarities in clustering methods.

In order to proceed with the critical assessment of the selected software, it was necessary to generate datasets that allowed fair comparisons of the different software. A systematic approach to the creation of synthetic datasets with controlled properties and data distributions to serve this purpose was achieved in Chapter 3.

Application of these synthetic datasets to assess performances of the automated soft-

ware was performed in Chapters 4 (for clusters containing different distances between clusters and with normal or skewed distributions), Chapter 5 (rare cells), and Chapter 6 (noise properties). The results presented in these chapters demonstrated the limitations and variability in outputs from automated systems.

To enable flow cytometry users to understand the relevance of these findings to both their own data and established data analysis workflows, the variability from software analysis of a clinical real-world dataset was compared to the variability from manual analysis by laboratory participants in Chapter 7.

This chapter here concludes the main findings of the thesis in the context of the initial research aims and objectives set out in Chapter 1. The central contributions of this research to existing knowledge are defined, and finally, upon reflection on the limitations of the thesis, recommendations for further work are proposed.

8.2 Thesis conclusions

Major findings from this thesis are listed below, each under a specific research aim and their corresponding research questions:

Research aim 1: Understand the landscape of automated data analysis software in the context of academic, clinical, and manufacturing laboratories.

- What automated flow cytometry data analysis software tools are available?
- How do the software tools differ?
- How widely used are the software tools?
- How do their usages differ between academic, clinical and manufacturing settings?

These research questions were addressed in the literature survey and clinical laboratory questionnaire completed in Chapter 2. The main findings from the literature survey were the availability of 51 software tools (at that time in 2019) that implement different supervised or unsupervised learning algorithms (dimensionality reduction, hierarchical clustering, model-based clustering, density-based clustering, etc.). The total number of citations from the 51 software tools was 2,027, and the ones with the highest citation rates were viSNE, PhenoGraph, SPADE1, FlowSOM and t-SNE.

In comparison, in the survey of clinical cytometrists, 16 respondents identified nine software tools actually used in their laboratories, with the most frequently software identified being Infinicyt, which uses a supervised learning algorithm for automated identification and classification of cell populations based on an extensive library of patient data.

It was revealing that the uptake of automated software among clinical laboratories was less than half (47%) of the 49 participants surveyed.

A comparison of the most frequently identified software from the two surveys suggested differences in usage purposes — those identified from literature could be considered tools for discovery or exploratory analyses, whereas those from clinical settings had more relevance in routine, targeted analyses of known cell populations.

Research aim 2: Explore benchmarking datasets applicable for the critical assessment of automated flow cytometry data analysis software.

- What properties of flow cytometry benchmarking datasets are required for testing of software tools?
- Can synthetic datasets be designed and generated with these properties?
- What are the advantages and disadvantages of synthetic datasets compared with real-world datasets?

The research completed in Chapter 3 identified common characteristics and statistical properties of flow cytometry data, such as the number of clusters, number of total events, number of markers, and cluster separation, placement and distribution. A method was developed using the R programming environment to computationally simulate certain targeted properties in a highly controlled manner. This approach successfully generated synthetic datasets with relevant cell population characteristics, in the standard FCS 3.1 file format, that could be processed in place of real-world data through software tools with credibility.

A driving force behind the use of synthetic datasets for external quality assessments of software was the shortcomings of real world flow cytometry data, which require estimates of a reference value (e.g. cell population frequencies) from manual analysis outputs of an individual or a group of experts. This reliance on manual gating of real world datasets, with its large source of variation, lack of reproducibility and susceptibility to bias was seen as a major disadvantage which the application of synthetic datasets could potentially overcome.

The key benefits of using synthetic datasets in this context are:

1. Ground truth designed into data means an absolute reference value can be used during testing, thereby moving away from using manually gated references, allowing a clear statement of accuracy.

2. A framework for synthetic data generation can be implemented to potentially create datasets matching any real-world flow cytometry dataset, in numerous cell models or disease types, and can overcome collection and ethical limitations of obtaining real samples from patients, for example.

3. Full reproducibility of the synthetic datasets by different laboratories. Since the code is based on mathematical principles, it can be run by other personnel with the same programming language knowledge to give identical outputs.

4. Isolation of the variation from data analysis from other sources of variation arising from sample processing, reagents and materials and instruments (inherent in real data), so that process improvement can be specifically targeted.

Although the main limitation of synthetic datasets lie in their accurate representation of the complexities of real data, the work in this thesis has shown how, through thoughtful designs that capture common key characteristics of real data, the synthetic datasets have demonstrated utility and credibility in giving users a better understanding of the clustering behaviours of different software, and the constraints on the types of data each software are able to analyse effectively. Additionally, the datasets inform selection of software to be used for desired end application (for instance, rare cell detection).

Research aim 3: Compare the performances of different software tools in cell population identification tasks.

- What is the effect of varying the distance between clusters on software performance (in terms of accuracy and repeatability)?
- Are certain software more sensitive to non-normal cluster distributions?
- What are the limits of detection of the different software when challenged with a rare cell dataset?
- How robust are software tools to noise elements in the data?

Extensive software comparison studies were performed to address these questions in Chapters 4 to 6. The key findings from these were the observations of different clustering characteristics, and considerable variation between different software when analysing the same dataset.

Chapter 4 set out to assess the accuracy and reproducibility of software using a synthetic dataset containing clusters with different degrees of separation along a separation index (SI). The comparison between different software tools found that all software displayed high accuracy and repeatability when clusters were well-separated, with a SI $\geq$

0. However, as the SI decreased below 0 and the clusters began to approach and overlap each other, all software performance deteriorated. For instance, the difference between FlowSOM's software output to the reference cell population percentage widened from $0.92\% \pm 0.35\%$ at a SI of $+0.1$ to $8.9\% \pm 11.4\%$ at a SI of -0.2. The results from these tests suggest that SWIFT was the best performing software for separating slightly overlapping clusters, because it had the lowest difference to reference value compared to the other software tools at SI values of -0.1 and -0.2. This finding was validated with the real world NEQAS dataset in Chapter 7, where SWIFT reported the lowest difference to reference amongst other software when tasked with identifying the CD34+ cell population, which was not a well-separated cluster.

The introduction of skewed clusters to the synthetic datasets found that SWIFT was most sensitive to these non-normal distributions, whereas FlowSOM, PhenoGraph, SPADE3 were less affected. Most noticeably, in the head-to-head orientated skewed dataset, SWIFT's difference to reference for clusters with heavy skew ($\alpha = 7.5$) deteriorated to $35.7\% \pm 21.6\%$ from a value of $2.6\% \pm 2.2\%$ with no skew ($\alpha = 0$). The negative impact of non-normal cluster distributions on SWIFT performance was further evidenced by its failure to identify the bead cluster from the NEQAS dataset in Chapter 7, as well as the synthetic rare-skew populations in Chapter 5.

Rare cell detection and analysis is a critical application in flow cytometry, and automated software tools have the potential to improve the quality of rare cell characterisations. The work completed in Chapter 5 found that the limits of detection (LoD) of rare cell populations varied between the different software. FlowSOM and SWIFT were the best performing software for detection of rare events, with both achieving a LoD of 500 cells in 10^6 total events (0.05%). In contrast, PhenoGraph and SPADE3 failed to detect any rare population below 5% in all total events. The LoD typically improved in percentage terms as the magnitude of total events increased, for instance, SWIFT detected a rare cell frequency of 5% at 10^3 total events, which lowered to 0.5% at both 10^4 and 10^5 total events, and lowered further to 0.05% at 10^6 total events.

The study in Chapter 5 also tested automated software with the task of identifying rare-skew populations, which revealed that SWIFT was unable to match its LoD from normally distributed clusters, as shown with its failure to identify any of the rare-skew populations in the 10^6 total events dataset. FlowSOM did not appear to be affected by the skewed clusters, and was the best performing software in this regard, managing to detect 100 rare-skew cells in 10^6 total events (0.01%). Unlike SWIFT, which is based on fitting data to Gaussian distributions in its clustering method, FlowSOM is based on

self-organising maps, and these differences in clustering strategy most probably explains their variation in responses to rare clusters with non-normal distributions.

The next area of investigation in Chapter 6 focussed on biological noise or outlier events that are ubiquitous in real data, but not considered in the previously generated clean synthetic datasets. The addition of noise events resulted in an expected decrease in performance from all software, compared to datasets with no noise. But interestingly, the different levels of noise distribution that were introduced (3SD and 4SD) brought about contrasting responses from the software. The deterioration in performance at the higher level of noise was most significant for FlowSOM, whereas PhenoGraph and SPADE3 showed no differences in performance. The results showed that software responses to noise elements were clearly mixed, and the individual levels of robustness displayed were most likely to be dependent on the underlying clustering algorithm utilised, along with its statistical distribution assumptions.

Research aim 4: Analyse the variation between automated software outputs in comparison to manually analysed data.

- How does the variation compare between manual and automated software analysis of flow cytometry data?
- Does the variation differ when analysing cell populations with different degrees of separation?
- What is the correlation between the two analysis methods?

The 'manual versus automated' comparison study in Chapter 7 found that for well-separated populations, automated software tools were able to outperform human operators with significantly lower coefficients of variation. Additionally, a strong correlation for cell counts was seen between the two methods when analysing distinct clusters. However, when clusters were closer together, many software gave unsatisfactory outputs and human intervention was required for quality control checks. Higher variability was associated with software outputs for these less well-separated clusters, and poor correlation between participants and software counts were seen. These findings indicate that confidence in automated software tools remains slightly short of that in manual gating methods, given that many flow cytometry datasets have close or touching cell populations, and achieving very well-separated populations are in practice limited by factors such as fluorophore brightness, background from non-specific staining, and levels of antigen expression.

8.3 Thesis novelty and contributions to knowledge

This thesis is perceived to have delivered the following novelties and contributions to knowledge:

1. Identification of the current trends in flow cytometry automated data analysis software from a literature survey.

2. Identification of real usage of automated software in clinical laboratories via a participant survey, which revealed low adoption rates, and differences between the most cited software in literature and the most used ones among clinical flow cytometry operators.

3. Presentation of a synthetic dataset approach to assess the performances of unsupervised learning automated flow cytometry data analysis software. This approach differs from previous work to critically assess flow cytometry computational tools, which have mostly used real-world datasets.

4. Development of a range of synthetic datasets with different flow cytometry data properties, including:

   (a) first use of the Separation Index in a flow cytometry setting to control the distances between clusters

   (b) novel application of a dataset with increasing levels of skew, in different orientation pairings

   (c) novel approach of using rare cell datasets with different magnitudes of total events, and also rare-skew cell populations, to test limits of detection of software

   (d) first use of a synthetic flow cytometry dataset with varying levels of noise distribution to assess software performance.

   Importantly, the properties simulated were tightly controlled, created with principles of Design of Experiment in mind, reproducible, and designed with a 'ground truth' not possible from real-world experimental data.

5. Application of these datasets to several software (Flock2, flowMeans, FlowSOM, PhenoGraph, SPADE1, SPADE3, and SWIFT), each of which employ a different class of clustering algorithm, to test their ability to identify target cell populations.

6. Quantification of the difference to reference values of these software outputs, as an indication of the measurement uncertainty arising from automated analyses.

7. Validation of this synthetic dataset approach by using real-world datasets with comparable data properties.

8. A demonstration of the shortcomings in the SWIFT algorithm for analysing non-normally distributed cell populations.

9. Illustration of the limitations in the detection of rare populations by PhenoGraph, SPADE1 and SPADE3.

10. A demonstration of the breakdown in FlowSOM performance when challenged with noise, leading to a better understanding of the importance of data cleaning and pre-processing prior to using analytical algorithms.

11. Comparison of inter-laboratory variation from manual analysis of a clinical dataset to variation from automated analysis. Analysis of outputs for well-separated cell populations showed strong correlation between manual and automated analyses, which suggests that software are able to reproduce manual gating in this particular task. However, a lack of readiness was evident for automated analysis of less well-separated populations.

8.4 Further work

8.4.1 Datasets

In this current work, only synthetic datasets with a limited number of clusters and dimensions have been generated. It is recognised that there is (at times) a disparity here to real world flow cytometry datasets that can be more complex, and include greater numbers of cell populations and more staining parameters (e.g. 6 to 10 colours). Related to this is also the issue where certain software, such as PhenoGraph, were not intended by their developers for analysis of such datasets with low numbers of populations and dimensions. So, while their performances at analysis of these lower-complexity datasets here were informative, it could be argued that they were not fully indicative of the software capability. Therefore, an obvious focus for further work would be to increase the complexity of synthetic datasets to better address these two issues.

A more advanced approach to optimise the features of benchmarking datasets is the potential use of data augmentation to create new data from existing real datasets, a

strategy that appears widely used in the field of image analysis [220], and has been demonstrated in imaging flow cytometry [221]. Strategies such as generative adversarial networks (GANs) could be explored to generate large scale training datasets and open up work in the evaluation of supervised learning algorithms [222].

Of potential benefit to flow cytometry data analysts is the concept of a library or repository of synthetic datasets, including the code that generates them, which users can search for, take off-the-shelf and apply to their novel data analysis workflows. Furthermore, a resource that can allow interactive analysis and interpretation of synthetic datasets by different software tools, on a cloud-based platform, would help users determine which automated software are most suited to their needs. This synthetic dataset repository could extend on previous work carried out on FlowRepository which provides publicly available, MIFlowCyt standard annotated experimental flow cytometry datasets [164].

8.4.2 Software

Seven software, each implementing a different type of clustering algorithm, were selected in the comparison studies performed here, as representatives of similar algorithms in their class. Further work could extend the range of unsupervised learning algorithms included, for example to investigate the variation of software within the same class of clustering algorithm. In line with this, the optimisation or tuning of individual software parameters was not investigated in this work due to timing constraints, however it could be an area for further investigation.

The scope of the main research in this thesis was focussed on software implementing unsupervised learning (specifically clustering) algorithms, because these appeared to be the most mature in their software development life cycle and were, in a way, commercially available. This leaves supervised learning software as a pertinent avenue for further research, albeit an area that would require substantial efforts in terms of dataset generation, curation, labelling; and software training and testing.

The nature of the benchmarking datasets generated in this work also necessitated the exclusion of dimensionality reduction algorithms, even though those were the most widely cited tools in the literature. Development of more complex synthetic datasets with more number of parameters may allow dimensionality reduction tools to be included in future software comparison studies.

Pre-processing software tools are increasingly being deployed to automatically clean, compensate, transform, and normalise cytometry data and prepare them for subsequent

advanced analyses. These tools that introduce modifications to the data are potentially a large source of variation which in turn have a marked impact on the integrity of downstream analyses. Research to assess the confidence in these tools is therefore warranted.

Finally, application of multiple automated software tools together in a pipeline has been documented [223, 84], however the combination of tools in the data analysis pipeline raises questions on the contributions to variability from each tool on the final output.

8.4.3 Wider work

- The analysis of inter-software variation compared to inter-laboratory variation presented in this thesis can be extended with further participant studies in collaboration with NEQAS. One line of inquiry would be to investigate variation from human participants when gating synthetic datasets (containing properties such as rare cells, noise) and to make comparisons with equivalent outputs from automated software analysis methods.
- Use synthetic datasets as an educational and training tool for operators, to review common manual gating pitfalls and areas for improvement, and to highlight areas of flow cytometry datasets that require careful analysis.
- Collaborate with bioindustry professionals and regulators to develop standards for use of automated software.
- Investigate other computational tools available for genomics or other -omics, mass cytometry, imaging software tools (e.g. imaging flow cytometry) that use algorithms for edge detection and segmentation. In particular, the wider landscape of automated tools available in diagnostics that fall under the umbrella of Software as a Medical Device among healthcare technologies.

8.5 Final remarks

Overall, the use of synthetic datasets in this thesis has highlighted many limitations in accuracy and reproducibility of current software when analysing challenging rare cells, ambiguously separated and irregularly shaped populations or more noisy flow cytometry data. These findings suggest it will be some time before automated data analysis tools can fully replace manual data analysis protocols, and therefore the role of human operators will remain critical in the characterisation process, but perhaps gradually shifted towards reviewal, adjustment and approval tasks. This research has shown how synthetic datasets are a valuable and agile toolset for evaluating measurement confidence from software

algorithms, and have the potential to be applied as digital reference materials to provide measurement assurances in automated characterisations of cell therapy products.

References

[1] Maecker HT, Rinfret A, D'Souza P, Darden J, Roig E, Landry C, et al. Standardization of cytokine flow cytometry assays. BMC Immunology. 2005;6:13.

[2] Gouttefangeas C, Chan C, Attig S, Køllgaard TT, Rammensee HG, Stevanović S, et al. Data analysis as a source of variability of the HLA-peptide multimer assay: from manual gating to automated recognition of cell clusters. Cancer Immunology, Immunotherapy. 2015;64(5):585-98.

[3] Pedersen NW, Chandran PA, Qian Y, Rebhahn J, Petersen NV, Hoff MD, et al. Automated analysis of flow cytometry data to reduce inter-lab variation in the detection of major histocompatibility complex multimer-binding T cells. Frontiers in Immunology. 2017;8:858.

[4] Keeney M, Hedley BD, Chin-Yee IH. Flow cytometry—Recognizing unusual populations in leukemia and lymphoma diagnosis. International Journal of Laboratory Hematology. 2017;39(S1):86-92.

[5] Cherian S, Hedley BD, Keeney M. Common flow cytometry pitfalls in diagnostic hematopathology. Cytometry Part B: Clinical Cytometry. 2019;96(6):449-63.

[6] Saeys Y, Van Gassen S, Lambrecht BN. Computational flow cytometry: helping to make sense of high-dimensional immunology data. Nature Reviews Immunology. 2016;16(7):449-62.

[7] Malik NN, Durdy MB. Chapter 7 - Cell Therapy Landscape: Autologous and Allogeneic Approaches. In: Atala A, Allickson JG, editors. Translational Regenerative Medicine. Boston: Academic Press; 2015. p. 87-106.

[8] Little MT, Storb R. History of haematopoietic stem-cell transplantation. Nature Reviews Cancer. 2002;2(3):231-8.

[9] Thomas ED, Lochte Jr HL, Lu WC, Ferrebee JW. Intravenous infusion of bone marrow in patients receiving radiation and chemotherapy. New England Journal of Medicine. 1957;257(11):491-6.

[10] Granot N, Storb R. History of hematopoietic cell transplantation: challenges and progress. Haematologica. 2020;105(12):2716-29.

[11] Passweg JR, Baldomero H, Chabannon C, Basak GW, de la Cámara R, Corbacioglu S, et al. Hematopoietic cell transplantation and cellular therapy survey of the EBMT: monitoring of activities and trends over 30 years. Bone Marrow Transplantation. 2021;56(7):1651-64.

[12] Phinney DG, Prockop DJ. Concise review: mesenchymal stem/multipotent stromal cells: the state of transdifferentiation and modes of tissue repair—current views. Stem Cells. 2007;25(11):2896-902.

[13] Levy O, Kuai R, Siren EM, Bhere D, Milton Y, Nissar N, et al. Shattering barriers toward clinically meaningful MSC therapies. Science Advances. 2020;6(30):eaba6884.

[14] Fesnak AD, June CH, Levine BL. Engineered T cells: the promise and challenges of cancer immunotherapy. Nature Reviews Cancer. 2016;16(9):566-81.

[15] Maude SL, Laetsch TW, Buechner J, Rives S, Boyer M, Bittencourt H, et al. Tisagenlecleucel in children and young adults with B-cell lymphoblastic leukemia. New England Journal of Medicine. 2018;378(5):439-48.

[16] Locke FL, Ghobadi A, Jacobson CA, Miklos DB, Lekakis LJ, Oluwole OO, et al. Long-term safety and activity of axicabtagene ciloleucel in refractory large B-cell lymphoma (ZUMA-1): a single-arm, multicentre, phase 1–2 trial. The Lancet Oncology. 2019;20(1):31-42.

[17] US Food and Drug Administration. Approved Cellular and Gene Therapy Products; 2021. (Date accessed 24 January 2022). Available from: https://www.fda.gov/vaccines-blood-biologics/cellular-gene-therapy-products/approved-cellular-and-gene-therapy-products.

[18] American Society of Gene & Cell Therapy. Gene, Cell, & RNA Therapy Landscape Q4 2021 Quarterly Data Report; 2022. (Date accessed 6 March 2022). Available from: https://asgct.org/global/documents/asgct-pharma-intelligence-quarterly-report-q4-2021.aspx.

[19] Advanced Therapies Manufacturing Taskforce. Advanced Therapies Manufacturing Action Plan: Retaining and attracting advanced therapies manufacture in the UK; 2016. (Date accessed 28 September 2018). Available from: https://www.bioindustry.org/resource-listing/advanced-therapies-manufacturing-

action-plan--retaining-and-attracting-advanced-therapies-manufacture-in-the-uk.html.

[20] Dodson BP, Levine AD. Challenges in the translation and commercialization of cell therapies. BMC Biotechnology. 2015;15:70.

[21] Levine BL, Miskin J, Wonnacott K, Keir C. Global manufacturing of CAR T cell therapy. Molecular Therapy-Methods & Clinical Development. 2017;4:92-101.

[22] Krasilnikova OA, Klabukov ID, Baranovskii DS, Shegay PV, Kaprin AD. The new legal framework for minimally manipulated cells expands the possibilities for cell therapy in Russia. Cytotherapy. 2021;23(8):754-5.

[23] Thurman-Newell JA, Petzing JN, Williams DJ. A meta-analysis of biological variation in blood-based therapy as a precursor to bio-manufacturing. Cytotherapy. 2016;18(5):686-94.

[24] Wang X, Rivière I. Clinical manufacturing of CAR T cells: foundation of a promising therapy. Molecular Therapy-Oncolytics. 2016;3:16015.

[25] Hollyman D, Stefanski J, Przybylowski M, Bartido S, Borquez-Ojeda O, Taylor C, et al. Manufacturing validation of biologically functional T cells targeted to CD19 antigen for autologous adoptive cell therapy. Journal of Immunotherapy. 2009;32(2):169-80.

[26] Reddy OL, Stroncek DF, Panch SR. Improving CAR T cell therapy by optimizing critical quality attributes. Seminars in Hematology. 2020;57(2):33-8.

[27] Kiesgen S, Messinger JC, Chintala NK, Tano Z, Adusumilli PS. Comparative analysis of assays to measure CAR T-cell-mediated cytotoxicity. Nature Protocols. 2021;16(3):1331-42.

[28] International Council for Harmonisation. ICH Q8 (R2) Pharmaceutical Development EMA/CHMP/ICH/167068/2004; 2009. (Date accessed 16 February 2022). Available from: https://www.ema.europa.eu/en/ich-q8-r2-pharmaceutical-development.

[29] International Council for Harmonisation. ICH Q9 Quality risk management EMA/CHMP/ICH/24235/2006; 2006. (Date accessed 16 February 2022). Available from: https://www.ema.europa.eu/en/ich-q9-quality-risk-management.

[30] International Council for Harmonisation. ICH Q10 Pharmaceutical quality system EMA/CHMP/ICH/214732/2007; 2008. (Date accessed 16 February 2022). Available from: https://www.ema.europa.eu/en/ich-q10-pharmaceutical-quality-system.

[31] European Medicines Agency. Guideline on quality, non-clinical and clinical requirements for investigational advanced therapy medicinal products in clinical trials EMA/CAT/852602/2018; 2019. (Date accessed 15 March 2022). Available from: https://www.ema.europa.eu/en/guideline-quality-non-clinical-clinical-requirements-investigational-advanced-therapy-medicinal.

[32] US Food and Drug Administration. Chemistry, Manufacturing, and Control (CMC) Information for Human Gene Therapy Investigational New Drug Applications (INDs) Guidance for Industry; 2020. (Date accessed 15 March 2022). Available from: https://www.fda.gov/regulatory-information/search-fda-guidance-documents/chemistry-manufacturing-and-control-cmc-information-human-gene-therapy-investigational-new-drug.

[33] Brown M, Wittwer C. Flow cytometry: principles and clinical applications in hematology. Clinical Chemistry. 2000;46(8):1221-9.

[34] Shapiro HM. Practical flow cytometry. 4th ed. Hoboken, NJ, USA: John Wiley & Sons; 2005.

[35] Chattopadhyay PK, Hogerkorp CM, Roederer M. A chromatic explosion: the development and future of multiparameter flow cytometry. Immunology. 2008;125(4):441-9.

[36] Mair F, Prlic M. OMIP-044: 28-color immunophenotyping of the human dendritic cell compartment. Cytometry Part A. 2018;93(4):402-5.

[37] Park LM, Lannigan J, Jaimes MC. OMIP-069: Forty-color full spectrum flow cytometry panel for deep immunophenotyping of major cell subsets in human peripheral blood. Cytometry Part A. 2020;97(10):1044-51.

[38] Spitzer MH, Nolan GP. Mass cytometry: single cells, many features. Cell. 2016;165(4):780-91.

[39] Sutherland DR, Anderson L, Keeney M, Nayar R, Chin-Yee I. The ISHAGE guidelines for CD34+ cell determination by flow cytometry. Journal of Hematotherapy. 1996;5(3):213-26.

[40] Keeney M, Chin-Yee I, Weir K, Popma J, Nayar R, Sutherland DR. Single platform flow cytometric absolute CD34+ cell counts based on the ISHAGE guidelines. Cytometry. 1998;34(2):61-70.

[41] Whitby A, Whitby L, Fletcher M, Reilly JT, Sutherland DR, Keeney M, et al. ISHAGE protocol: Are we doing it correctly? Cytometry Part B: Clinical Cytometry. 2012;82B(1):9-17.

[42] Grant R, Coopman K, Medcalf N, Silva-Gomes S, Campbell JJ, Kara B, et al. Understanding the contribution of operator measurement variability within flow cytometry data analysis for quality control of cell and gene therapy manufacturing. Measurement. 2020;150:106998.

[43] Grant R, Coopman K, Medcalf N, Silva-Gomes S, Campbell JJ, Kara B, et al. Quantifying operator subjectivity within flow cytometry data analysis as a source of measurement uncertainty and the impact of experience on results. PDA Journal of Pharmaceutical Science and Technology. 2021;75(1):33-47.

[44] Bashashati A, Brinkman RR. A survey of flow cytometry data analysis methods. Advances in Bioinformatics. 2009;2009:584603.

[45] Chester C, Maecker HT. Algorithmic tools for mining high-dimensional cytometry data. The Journal of Immunology. 2015;195(3):773-9.

[46] Mair F, Hartmann FJ, Mrdjen D, Tosevski V, Krieg C, Becher B. The end of gating? An introduction to automated analysis of high dimensional cytometry data. European Journal of Immunology. 2016;46(1):34-43.

[47] Weber LM, Robinson MD. Comparison of clustering methods for high-dimensional single-cell flow and mass cytometry data. Cytometry Part A. 2016;89(12):1084-96.

[48] Kimball AK, Oko LM, Bullock BL, Nemenoff RA, van Dyk LF, Clambey ET. A beginner's guide to analyzing and visualizing mass cytometry data. The Journal of Immunology. 2018;200(1):3-22.

[49] Aghaeepour N, Finak G, Hoos H, Mosmann TR, Brinkman R, Gottardo R, et al. Critical assessment of automated flow cytometry data analysis techniques. Nature Methods. 2013;10(3):228-38.

[50] Lee JA, Spidlen J, Boyce K, Cai J, Crosbie N, Dalphin M, et al. MIFlowCyt: The minimum information about a flow cytometry experiment. Cytometry Part A. 2008;73A(10):926-30.

[51] Spidlen J, Moore W, Parks D, Goldberg M, Bray C, Bierre P, et al. Data File Standard for Flow Cytometry, version FCS 3.1. Cytometry Part A. 2010;77A(1):97-100.

[52] Spidlen J, Leif RC, Moore W, Roederer M, Brinkman RR. Gating-ML: XML-based gating descriptions in flow cytometry. Cytometry Part A. 2008;73A(12):1151-7.

[53] Kalina T, Flores-Montero J, Van Der Velden V, Martin-Ayuso M, Böttcher S, Ritgen M, et al. EuroFlow standardization of flow cytometer instrument settings and immunophenotyping protocols. Leukemia. 2012;26(9):1986-2010.

[54] Finak G, Langweiler M, Jaimes M, Malek M, Taghiyar J, Korin Y, et al. Standardizing flow cytometry immunophenotyping analysis from the human immunophenotyping consortium. Scientific Reports. 2016;6:20686.

[55] International Clinical Cytometry Society. ICCS Quality & Standards Committee; 2020. (Date accessed 23 November 2020). Available from: https://www.cytometry.org/web/quality.php.

[56] Clinical & Laboratory Standards Institute. CLSI Guidelines; 2020. (Date accessed 23 November 2020). Available from: https://clsi.org/.

[57] Whitby L, Whitby A, Fletcher M, Barnett D. Current laboratory practices in flow cytometry for the enumeration of CD 4+ T-lymphocyte subsets. Cytometry Part B: Clinical Cytometry. 2015;88(5):305-11.

[58] Stebbings R, Wang L, Sutherland J, Kammel M, Gaigalas AK, John M, et al. Quantification of cells with specific phenotypes I: Determination of CD4+ cell count per microliter in reconstituted lyophilized human PBMC prelabeled with anti-CD4 FITC antibody. Cytometry Part A. 2015;87(3):244-53.

[59] Bendall SC, Nolan GP, Roederer M, Chattopadhyay PK. A deep profiler's guide to cytometry. Trends in Immunology. 2012;33(7):323-32.

[60] Bendall SC, Simonds EF, Qiu P, Amir EaD, Krutzik PO, Finck R, et al. Single-cell mass cytometry of differential immune and drug responses across a human hematopoietic continuum. Science. 2011;332(6030):687-96.

[61] O'Neill K, Aghaeepour N, Špidlen J, Brinkman R. Flow cytometry bioinformatics. PLoS Computational Biology. 2013;9(12):e1003365.

[62] Aghaeepour N, Chattopadhyay P, Chikina M, Dhaene T, Van Gassen S, Kursa M, et al. A benchmark for evaluation of algorithms for identification of cellular correlates of clinical outcomes. Cytometry Part A. 2016;89(1):16-21.

[63] Melchiotti R, Gracio F, Kordasti S, Todd AK, de Rinaldis E. Cluster stability in the analysis of mass cytometry data. Cytometry Part A. 2017;91(1):73-84.

[64] Czechowska K, Lannigan J, Wang L, Arcidiacono J, Ashhurst TM, Barnard RM, et al. Cyt-Geist: Current and future challenges in cytometry: Reports of the CYTO 2018 Conference Workshops. Cytometry Part A. 2019;95(6):598-644.

[65] Zunder ER, Lujan E, Goltsev Y, Wernig M, Nolan GP. A continuous molecular roadmap to iPSC reprogramming through progression analysis of single-cell mass cytometry. Cell Stem Cell. 2015;16(3):323-37.

[66] Spitzer MH, Gherardini PF, Fragiadakis GK, Bhattacharya N, Yuan RT, Hotson AN, et al. An interactive reference framework for modeling a dynamic immune system. Science. 2015;349(6244):1259425.

[67] Amir EaD, Davis KL, Tadmor MD, Simonds EF, Levine JH, Bendall SC, et al. viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. Nature Biotechnology. 2013;31(6):545-52.

[68] Qiu P, Simonds EF, Bendall SC, Gibbs KD, Bruggner RV, Linderman MD, et al. Extracting a cellular hierarchy from high-dimensional cytometry data with SPADE. Nature Biotechnology. 2011;29(10):886-93.

[69] van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of Machine Learning Research. 2008;9(86):2579-605.

[70] Levine JH, Simonds EF, Bendall SC, Davis KL, Amir EAD, Tadmor MD, et al. Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis. Cell. 2015;162(1):184-97.

[71] Pyne S, Hu X, Wang K, Rossin E, Lin TI, Maier LM, et al. Automated high-dimensional flow cytometric data analysis. Proceedings of the National Academy of Sciences. 2009;106(21):8519-24.

[72] Bruggner RV, Bodenmiller B, Dill DL, Tibshirani RJ, Nolan GP. Automated identification of stratifying signatures in cellular subpopulations. Proceedings of the National Academy of Sciences. 2014;111(26):E2770-7.

[73] van Gassen S, Callebaut B, Van Helden MJ, Lambrecht BN, Demeester P, Dhaene T, et al. FlowSOM: Using self-organizing maps for visualization and interpretation of cytometry data. Cytometry Part A. 2015;87(7):636-45.

[74] Becher B, Schlitzer A, Chen J, Mair F, Sumatoh HR, Teng KWW, et al. High-dimensional analysis of the murine myeloid cell system. Nature Immunology. 2014;15(12):1181-9.

[75] Aghaeepour N, Nikolic R, Hoos HH, Brinkman RR. Rapid cell population identification in flow cytometry data. Cytometry Part A. 2011;79 A(1):6-13.

[76] Shekhar K, Brodin P, Davis MM, Chakraborty AK. Automatic classification of cellular expression by nonlinear stochastic embedding (ACCENSE). Proceedings of the National Academy of Sciences. 2014;111(1):202-7.

[77] Bendall SC, Davis KL, Amir EaD, Tadmor MD, Simonds EF, Chen TJ, et al. Single-cell trajectory detection uncovers progression and regulatory coordination in human B cell development. Cell. 2014;157(3):714-25.

[78] Qian Y, Wei C, Eun-Hyung Lee F, Campbell J, Halliley J, Lee JA, et al. Elucidation of seventeen human peripheral blood B-cell subsets and quantification of the tetanus response using a density-based method for the automated identification of cell populations in multidimensional flow cytometry data. Cytometry Part B: Clinical Cytometry. 2010;78B(S1):S69-82.

[79] Lo K, Hahne F, Brinkman RR, Gottardo R. flowClust: a Bioconductor package for automated gating of flow cytometry data. BMC Bioinformatics. 2009;10:145.

[80] Finak G, Bashashati A, Brinkman R, Gottardo R. Merging mixture components for cell population identification in flow cytometry. Advances in Bioinformatics. 2009;2009:247646.

[81] Samusik N, Good Z, Spitzer MH, Davis KL, Nolan GP. Automated mapping of phenotype space with single-cell data. Nature Methods. 2016;13(6):493-6.

[82] Zare H, Shooshtari P, Gupta A, Brinkman RR. Data reduction for spectral clustering to analyze high throughput flow cytometry data. BMC Bioinformatics. 2010;11:403.

[83] Ge Y, Sealfon SC. Flowpeaks: A fast unsupervised clustering for flow cytometry data via K-means and density peak finding. Bioinformatics. 2012;28(15):2052-8.

[84] Finak G, Frelinger J, Jiang W, Newell EW, Ramey J, Davis MM, et al. OpenCyto: an open source infrastructure for scalable, robust, reproducible, and automated, end-to-end flow cytometry data analysis. PLoS Computational Biology. 2014;10(8):e1003806.

[85] Boedigheimer MJ, Ferbas J. Mixture modeling approach to flow cytometry data. Cytometry Part A. 2008;73(5):421-9.

[86] Cron A, Gouttefangeas C, Frelinger J, Lin L, Singh SK, Britten CM, et al. Hierarchical modeling for rare event detection and cell subset alignment across flow cytometry samples. PLoS Computational Biology. 2013;9(7):e1003130.

[87] Malek M, Taghiyar MJ, Chong L, Finak G, Gottardo R, Brinkman RR. flowDensity: reproducing manual gating of flow cytometry data by automated density-based cell population identification. Bioinformatics. 2015;31(4):606-7.

[88] Naim I, Datta S, Rebhahn J, Cavenaugh JS, Mosmann TR, Sharma G. SWIFT-scalable clustering for automated identification of rare cell populations in large, high-dimensional flow cytometry datasets, Part 1: Algorithm design. Cytometry Part A. 2014;85(5):408-21.

[89] Mosmann TR, Naim I, Rebhahn J, Datta S, Cavenaugh JS, Weaver JM, et al. SWIFT-scalable clustering for automated identification of rare cell populations in large, high-dimensional flow cytometry datasets, Part 2: Biological evaluation. Cytometry Part A. 2014;85(5):422-33.

[90] Diggins KE, Ferrell Jr PB, Irish JM. Methods for discovery and characterization of cell subsets in high dimensional mass cytometry data. Methods. 2015;82:55-63.

[91] Sugar IP, Sealfon SC. Misty Mountain clustering: application to fast unsupervised flow cytometry gating. BMC Bioinformatics. 2010;11:502.

[92] Lin L, Finak G, Ushey K, Seshadri C, Hawn TR, Frahm N, et al. COMPASS identifies T-cell subsets correlated with clinical outcomes. Nature Biotechnology. 2015;33(6):610-6.

[93] Rogers WT, Moser AR, Holyst HA, Bantly A, Mohler III ER, Scangas G, et al. Cytometric fingerprinting: quantitative characterization of multivariate distributions. Cytometry Part A. 2008;73(5):430-41.

[94] Sörensen T, Baumgart S, Durek P, Grützkau A, Häupl T. immunoClust—An automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. Cytometry Part A. 2015;87(7):603-15.

[95] Pyne S, Lee SX, Wang K, Irish J, Tamayo P, Nazaire MD, et al. Joint modeling and registration of cell populations in cohorts of high-dimensional flow cytometric data. PloS One. 2014;9(7):e100334.

[96] O'Neill K, Jalali A, Aghaeepour N, Hoos H, Brinkman RR. Enhanced flowType/RchyOptimyx: a BioConductor pipeline for discovery in high-dimensional cytometry data. Bioinformatics. 2014;30(9):1329-30.

[97] Dundar M, Akova F, Yerebakan HZ, Rajwa B. A non-parametric Bayesian model for joint cell clustering and cluster matching: identification of anomalous sample phenotypes with random effects. BMC Bioinformatics. 2014;15:314.

[98] Li H, Shaham U, Stanton KP, Yao Y, Montgomery RR, Kluger Y. Gating mass cytometry data by deep learning. Bioinformatics. 2017;33(21):3423-30.

[99] Meehan S, Walther G, Moore W, Orlova D, Meehan C, Parks D, et al. AutoGate: automating analysis of flow cytometry data. Immunologic Research. 2014;58(2):218-23.

[100] van Gassen S, Vens C, Dhaene T, Lambrecht BN, Saeys Y. FloReMi: Flow density survival regression using minimal feature redundancy. Cytometry Part A. 2016;89(1):22-9.

[101] Anchang B, Do MT, Zhao X, Plevritis SK. CCAST: a model-based gating strategy to isolate homogeneous subpopulations in a heterogeneous population of single cells. PLoS Computational Biology. 2014;10(7):e1003664.

[102] Lux M, Brinkman RR, Chauve C, Laing A, Lorenc A, Abeler-Dörner L, et al. flowLearn: fast and precise identification and quality checking of cell populations in flow cytometry. Bioinformatics. 2018;34(13):2245-53.

[103] Lee HC, Kosoy R, Becker CE, Dudley JT, Kidd BA. Automated cell type discovery and classification through knowledge transfer. Bioinformatics. 2017;33(11):1689-95.

[104] Rebhahn JA, Roumanes DR, Qi Y, Khan A, Thakar J, Rosenberg A, et al. Competitive SWIFT cluster templates enhance detection of aging changes. Cytometry Part A. 2016;89(1):59-70.

[105] Qiu P. Toward deterministic and semiautomated SPADE analysis. Cytometry Part A. 2017;91(3):281-9.

[106] Commenges D, Alkhassim C, Gottardo R, Hejblum B, Thiébaut R. cytometree: A binary tree algorithm for automatic gating in cytometry analysis. Cytometry Part A. 2018;93(11):1132-40.

[107] Lee AJ, Chang I, Burel JG, Lindestam Arlehamn CS, Mandava A, Weiskopf D, et al. DAFi: A directed recursive data filtering and clustering approach for improving and interpreting data clustering identification of cell populations from polychromatic flow cytometry data. Cytometry Part A. 2018;93(6):597-610.

[108] Weber LM, Nowicka M, Soneson C, Robinson MD. diffcyt: Differential discovery in high-dimensional cytometry via high-resolution clustering. Communications Biology. 2019;2:183.

[109] Reiter M, Rota P, Kleber F, Diem M, Groeneveld-Krentz S, Dworzak M. Clustering of cell populations in flow cytometry data using a combination of Gaussian mixtures. Pattern Recognition. 2016;60:1029-40.

[110] Abdelaal T, van Unen V, Höllt T, Koning F, Reinders MJ, Mahfouz A. Predicting cell populations in single cell mass cytometry data. Cytometry Part A. 2019;95(7):769-81.

[111] Folcarelli R, van Staveren S, Bouman R, Hilvering B, Tinnevelt GH, Postma G, et al. Automated flow cytometric identification of disease-specific cells by the ECLIPSE algorithm. Scientific Reports. 2018;8:10907.

[112] Hejblum BP, Alkhassim C, Gottardo R, Caron F, Thiébaut R. Sequential Dirichlet process mixtures of multivariate skew t-distributions for model-based clustering of flow cytometry data. The Annals of Applied Statistics. 2019;13(1):638-60.

[113] Wong L, Hill BL, Hunsberger BC, Bagwell CB, Curtis AD, Davis BH. Automated analysis of flow cytometric data for measuring neutrophil CD64 expression using a multi-instrument compatible probability state model. Cytometry Part B: Clinical Cytometry. 2015;88(4):227-35.

[114] Zaunders J, Jing J, Leipold M, Maecker H, Kelleher AD, Koch I. Computationally efficient multidimensional analysis of complex flow cytometry data using second order polynomial histograms. Cytometry Part A. 2016;89(1):44-58.

[115] Moon KR, van Dijk D, Wang Z, Gigante S, Burkhardt DB, Chen WS, et al. Visualizing structure and transitions in high-dimensional biological data. Nature Biotechnology. 2019;37(12):1482-92.

[116] Meehan S, Kolyagin GA, Parks D, Youngyunpipatkul J, Herzenberg LA, Walther G, et al. Automated subset identification and characterization pipeline for multidimensional flow and mass cytometry data clustering and visualization. Communications Biology. 2019;2:229.

[117] Pouyan MB, Nourani M. Identifying cell populations in flow cytometry data using phenotypic signatures. IEEE/ACM Transactions on Computational Biology and Bioinformatics. 2016;14(4):880-91.

[118] Becht E, McInnes L, Healy J, Dutertre CA, Kwok IW, Ng LG, et al. Dimensionality reduction for visualizing single-cell data using UMAP. Nature Biotechnology. 2019;37(1):38-44.

[119] Kotecha N, Krutzik PO, Irish JM. Web-based analysis and publication of flow cytometry experiments. Current Protocols in Cytometry. 2010;53(1):10-7.

[120] Xu D, Tian Y. A comprehensive survey of clustering algorithms. Annals of Data Science. 2015;2(2):165-93.

[121] Kaufman L, Rousseeuw PJ. Finding groups in data: an introduction to cluster analysis. John Wiley & Sons; 2009.

[122] Jain AK. Data clustering: 50 years beyond K-means. Pattern Recognition Letters. 2010;31(8):651-66.

[123] Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment. 2008;2008(10):P10008.

[124] Ester M, Kriegel HP, Sander J, Xu X, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: kdd. vol. 96; 1996. p. 226-31.

[125] Ankerst M, Breunig MM, Kriegel HP, Sander J. OPTICS: ordering points to identify the clustering structure. ACM Sigmod Record. 1999;28(2):49-60.

[126] Fraley C, Raftery AE. Model-based clustering, discriminant analysis, and density estimation. Journal of the American Statistical Association. 2002;97(458):611-31.

[127] von Luxburg U. A tutorial on spectral clustering. Statistics and Computing. 2007;17(4):395-416.

[128] Kohonen T. The self-organizing map. Proceedings of the IEEE. 1990;78(9):1464-80.

[129] Kohonen T. Essentials of the self-organizing map. Neural Networks. 2013;37:52-65.

[130] Vesanto J, Alhoniemi E. Clustering of the self-organizing map. IEEE Transactions on Neural Networks. 2000;11(3):586-600.

[131] van Unen V, Höllt T, Pezzotti N, Li N, Reinders MJ, Eisemann E, et al. Visual analysis of mass cytometry data by hierarchical stochastic neighbour embedding reveals rare cell types. Nature Communications. 2017;8:1740.

[132] Fletez-Brant K, Špidlen J, Brinkman RR, Roederer M, Chattopadhyay PK. flowClean: Automated identification and removal of fluorescence anomalies in flow cytometry data. Cytometry Part A. 2016;89(5):461-71.

[133] Monaco G, Chen H, Poidinger M, Chen J, de Magalhães JP, Larbi A. flowAI: automatic and interactive anomaly discerning tools for flow cytometry data. Bioinformatics. 2016;32(16):2473-80.

[134] Hahne F, LeMeur N, Brinkman RR, Ellis B, Haaland P, Sarkar D, et al. flowCore: a Bioconductor package for high throughput flow cytometry. BMC Bioinformatics. 2009;10:106.

[135] Hahne F, Gopalakrishnan N, Khodabakhshi AH, Wong C, Lee K. flowStats: Statistical methods for the analysis of flow cytometry data; 2009. R Package version 3.1.

[136] Finak G, Jiang M. flowWorkspace: Infrastructure for representing and interacting with gated and ungated cytometry data sets; 2018. R package version 3.0.

[137] Linderman GC, Rachh M, Hoskins JG, Steinerberger S, Kluger Y. Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data. Nature Methods. 2019;16(3):243-5.

[138] Belkina AC, Ciccolella CO, Anno R, Halpert R, Spidlen J, Snyder-Cappione JE. Automated optimized parameters for T-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. Nature Communications. 2019;10:5415.

[139] Häkkinen A, Koiranen J, Casado J, Kaipio K, Lehtonen O, Petrucci E, et al. qSNE: quadratic rate t-SNE optimizer with automatic parameter tuning for large datasets. Bioinformatics. 2020;36(20):5086-92.

[140] Becht E, Tolstrup D, Dutertre CA, Morawski PA, Campbell DJ, Ginhoux F, et al. High-throughput single-cell quantification of hundreds of proteins using conventional flow cytometry and machine learning. Science Advances. 2021;7(39):eabg0505.

[141] Tinnevelt GH, van Staveren S, Wouters K, Wijnands E, Verboven K, Folcarelli R, et al. A novel data fusion method for the effective analysis of multiple panels of flow cytometry data. Scientific Reports. 2019;9:6777.

[142] Greene E, Finak G, D'Amico LA, Bhardwaj N, Church CD, Morishima C, et al. New interpretable machine-learning method for single-cell data reveals correlates of clinical response to cancer immunotherapy. Patterns. 2021;2:100372.

[143] Roca CP, Burton OT, Gergelits V, Prezzemolo T, Whyte CE, Halpert R, et al. AutoSpill is a principled framework that simplifies the analysis of multichromatic flow cytometry data. Nature Communications. 2021;12:2890.

[144] UK NEQAS LI. UK NEQAS LI EQA/PT Programmes; 2022. (Date accessed 28 March 2022). Available from: http://www.ukneqasli.co.uk/eqa-pt-programmes/.

[145] Pedersen MJ, Nielsen CV. Improving survey response rates in online panels: Effects of low-cost incentives and cost-free text appeal interventions. Social Science Computer Review. 2016;34(2):229-43.

[146] Brooke J. SUS: A 'Quick and Dirty' Usability Scale. In: Jordan PW, Thomas B, McClelland IL, Weerdmeester B, editors. Usability Evaluation in Industry. vol. 189. London: Taylor & Francis Ltd; 1996. p. 189-94.

[147] Bangor A, Kortum PT, Miller JT. An empirical evaluation of the system usability scale. International Journal of Human–Computer Interaction. 2008;24(6):574-94.

[148] Costa E, Pedreira CE, Barrena S, Lecrevisse Q, Flores J, Quijano S, et al. Automated pattern-guided principal component analysis vs expert-based immunophenotypic classification of B-cell chronic lymphoproliferative disorders: a step forward in the standardization of clinical immunophenotyping. Leukemia. 2010;24(11):1927-33.

[149] Lhermitte L, Mejstrikova E, Van Der Sluijs-Gelling A, Grigore GE, Sedek L, Bras A, et al. Automated database-guided expert-supervised orientation for immunophenotypic diagnosis and classification of acute leukemia. Leukemia. 2018;32(4):874-81.

[150] van Dongen J, Lhermitte L, Böttcher S, Almeida J, Van Der Velden V, Flores-Montero J, et al. EuroFlow antibody panels for standardized n-dimensional flow cytometric immunophenotyping of normal, reactive and malignant leukocytes. Leukemia. 2012;26(9):1908-75.

[151] International Organization for Standardization. ISO 15189:2012 Medical laboratories — Requirements for quality and competence. Geneva, Switzerland: ISO; 2012.

[152] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research. 2011;12(Oct):2825-30.

[153] Ros G, Sellart L, Materzynska J, Vazquez D, Lopez AM. The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2016:3234-43.

[154] Wrenninge M, Unger J. Synscapes: A photorealistic synthetic dataset for street scene parsing. arXiv [Preprint]. 2018. (Date accessed 20 November 2020). Available from: https://arxiv.org/abs/1810.08705.

[155] Tremblay J, To T, Birchfield S. Falling things: A synthetic dataset for 3D object detection and pose estimation. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. 2018;2018-June:2119-22.

[156] Hagiwara A, Warntjes M, Hori M, Andica C, Nakazawa M, Kumamaru KK, et al. SyMRI of the brain: Rapid quantification of relaxation rates and proton density, with synthetic MRI, automatic brain segmentation, and myelin measurement. Investigative Radiology. 2017;52(10):647-57.

[157] Ratanaprasatporn L, Chikarmane SA, Giess CS. Strengths and weaknesses of synthetic mammography in screening. Radiographics. 2017;37(7):1913-27.

[158] Niazi MKK, Parwani AV, Gurcan MN. Digital pathology and artificial intelligence. The Lancet Oncology. 2019;20(5):e253-61.

[159] Perez L, Wang J. The effectiveness of data augmentation in image classification using deep learning. arXiv [Preprint]. 2017. (Date accessed 5 January 2021). Available from: https://arxiv.org/abs/1712.04621.

[160] Arvaniti E, Claassen M. Sensitive detection of rare disease-associated cell subsets via representation learning. Nature Communications. 2017;8:14825.

[161] The European Parliament and the Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). Official Journal of the European Union. 2016;L119:16-32.

[162] Bigos M. Separation index: An easy-to-use metric for evaluation of different configurations on the same flow cytometer. Current Protocols in Cytometry. 2007;40(1):1-21.

[163] Telford WG, Babin SA, Khorev SV, Rowe SH. Green fiber lasers: An alternative to traditional DPSS green lasers for flow cytometry. Cytometry Part A. 2009;75(12):1031-9.

[164] Spidlen J, Breuer K, Rosenberg C, Kotecha N, Brinkman RR. FlowRepository: A resource of annotated flow cytometry datasets associated with peer-reviewed publications. Cytometry Part A. 2012;81(9):727-31.

[165] International Organization for Standardization. ISO 13528:2005 Statistical methods for use in proficiency testing by interlaboratory comparison. Geneva, Switzerland: ISO; 2005.

[166] Wang L, Abbasi F, Ornatsky O, Cole KD, Misakian M, Gaigalas AK, et al. Human CD4 + lymphocytes for antigen quantification: Characterization using conventional flow cytometry and mass cytometry. Cytometry Part A. 2012;81 A(7):567-75.

[167] Cheung M, Campbell JJ, Whitby L, Thomas RJ, Braybrook J, Petzing J. Current trends in flow cytometry automated data analysis software. Cytometry Part A. 2021;99(10):1007-21.

[168] Hadley W. Tidy data. The Journal of Statistical Software. 2014;59(1).

[169] Kotz S, Balakrishnan N, Johnson NL. Continuous multivariate distributions, Volume 1: Models and applications. vol. 1. John Wiley & Sons; 2004.

[170] Roederer M. Spectral compensation for flow cytometry: visualization artifacts, limitations, and caveats. Cytometry. 2001;45(3):194-205.

[171] Qiu W, Joe H. Separation index and partial membership for clustering. Computational Statistics and Data Analysis. 2006;50(3):585-603.

[172] Qiu W, Joe H. clusterGeneration: Random cluster generation (with specified degree of separation); 2020. R package version 1.3.5.

[173] Azzalini A. The R package 'sn': The skew-normal and related distributions such as the skew-t. Università di Padova, Italia; 2020. R package version 1.6-2.

[174] Wickham H, Seidel D. scales: Scale functions for visualization; 2020. R package version 1.1.1.

[175] Revelle W. psych: Procedures for psychological, psychometric, and personality research. Evanston, Illinois; 2020. R package version 2.0.12.

[176] Fleiss JL, Levin B, Paik MC. Statistical Methods for Rates and Proportions. 3rd ed. Hoboken, NJ, USA: John Wiley & Sons; 2003.

[177] Tharwat A. Classification assessment methods. Applied Computing and Informatics. 2020;17(1):168-92.

[178] Burel JG, Qian Y, Lindestam Arlehamn C, Weiskopf D, Zapardiel-Gonzalo J, Taplitz R, et al. An integrated workflow to assess technical and biological variability of cell population frequencies in human peripheral blood by flow cytometry. The Journal of Immunology. 2017;198(4):1748-58.

[179] Salati S, Zini R, Bianchi E, Testa A, Mavilio F, Manfredini R, et al. Role of CD34 antigen in myeloid differentiation of human hematopoietic progenitor cells. Stem Cells. 2008;26(4):950-9.

[180] Joanes DN, Gill CA. Comparing measures of sample skewness and kurtosis. Journal of the Royal Statistical Society: Series D (The Statistician). 1998;47(1):183-9.

[181] Azzalini A, Capitanio A. Statistical applications of the multivariate skew normal distribution. Journal of the Royal Statistical Society Series B: Statistical Methodology. 1999;61(3):579-602.

[182] Bhattacharya S, Dunn P, Thomas CG, Smith B, Schaefer H, Chen J, et al. ImmPort, toward repurposing of open access immunological assay data for translational and clinical research. Scientific Data. 2018;5:1-9.

[183] Chen H. Rphenograph: R implementation of the phenograph algorithm; 2015. R package version 0.99.1.

[184] Lorimer T, Held J, Stoop R. Clustering: how much bias do we need? Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. 2017;375(2096):20160293.

[185] Demaret J, Varlet P, Trauet J, Beauvais D, Grossemy A, Hégo F, et al. Monitoring CAR T-cells using flow cytometry. Cytometry Part B: Clinical Cytometry. 2021;100(2):218-24.

[186] Grant R, Coopman K, Silva-Gomes S, Campbell JJ, Kara B, Braybrook J, et al. Assessment of protocol impact on subjectivity uncertainty when analyzing peripheral blood mononuclear cell flow cytometry data files. Methods and Protocols. 2021;4(2):24.

[187] Donnenberg AD, Donnenberg VS. Rare-event analysis in flow cytometry. Clinics in Laboratory Medicine. 2007;27(3):627-52.

[188] Suda T, Suda J, Ogawa M. Single-cell origin of mouse hemopoietic colonies expressing multiple lineages in variable combinations. Proceedings of the National Academy of Sciences. 1983;80(21):6689-93.

[189] Huang S, Terstappen LW. Formation of haematopoietic microenvironment and haematopoietic stem cells from single human bone marrow stem cells. Nature. 1992;360(6406):745-9.

[190] Copelan EA. Hematopoietic stem-cell transplantation. New England Journal of Medicine. 2006;354(17):1813-26.

[191] US Food and Drug Administration. Hematologic malignancies: regulatory considerations for use of minimal residual disease in development of drug and biological products for treatment - Guidance for industry; 2020. (Date accessed 27 August 2021). Available from: https://www.fda.gov/regulatory-information/search-fda-guidance-documents/hematologic-malignancies-regulatory-considerations-use-minimal-residual-disease-development-drug-and.

[192] European Medicines Agency. Guideline on the use of minimal residual disease as a clinical endpoint in multiple myeloma studies; 2018. (Date accessed 27 August 2021). Available from: https://www.ema.europa.eu/en/guideline-use-minimal-residual-disease-clinical-endpoint-multiple-myeloma-studies.

[193] Allan AL, Keeney M. Circulating tumor cell analysis: technical and statistical considerations for application to the clinic. Journal of Oncology. 2010;2010:1-10.

[194] Xu Y, Zhang M, Ramos CA, Durett A, Liu E, Dakhova O, et al. Closely related T-memory stem cells correlate with in vivo expansion of CAR.CD19-T cells and are preserved by IL-7 and IL-15. Blood. 2014;123(24):3750-9.

[195] Armbruster DA, Pry T. Limit of blank, limit of detection and limit of quantitation. The Clinical Biochemist Reviews. 2008;29(Suppl 1):S49-52.

[196] International Organization for Standardization. ISO 5725-1:1994 Accuracy (trueness and precision) of measurement methods and results — Part 1: General principles and definitions. Geneva, Switzerland: ISO; 1994.

[197] Wood B, Jevremovic D, Béné MC, Yan M, Jacobs P, Litwin V, et al. Validation of cell-based fluorescence assays: Practice guidelines from the ICSH and ICCS–part V–assay performance criteria. Cytometry Part B: Clinical Cytometry. 2013;84(5):315-23.

[198] Zimmerlin L, Donnenberg VS, Donnenberg AD. In: Hawley TS, Hawley RG, editors. Rare Event Detection and Analysis in Flow Cytometry: Bone Marrow Mesenchymal Stem Cells, Breast Cancer Stem/Progenitor Cells in Malignant Effusions, and Pericytes in Disaggregated Adipose Tissue. Totowa, NJ: Humana Press; 2011. p. 251-73.

[199] Hedley B, Keeney M. Technical issues: flow cytometry and rare event analysis. International Journal of Laboratory Hematology. 2013;35(3):344-50.

[200] Arroz M, Came N, Lin P, Chen W, Yuan C, Lagoo A, et al. Consensus guidelines on plasma cell myeloma minimal residual disease analysis and reporting. Cytometry Part B: Clinical Cytometry. 2016;90(1):31-9.

[201] Lee H, Sun Y, Patti-Diaz L, Hedrick M, Ehrhardt AG. High-throughput analysis of clinical flow cytometry data by automated gating. Bioinformatics and Biology Insights. 2019;13:1177932219838851.

[202] ATCC. K-562 CCL-243 Product sheet; 2021. (Date accessed 29 September 2021). Available from: https://www.atcc.org/products/ccl-243.

[203] ATCC. K-562-GFP CCL-243-GFP Product sheet; 2021. (Date accessed 29 September 2021). Available from: https://www.atcc.org/products/ccl-243-gfp.

[204] Orlova DY, Meehan S, Parks D, Moore WA, Meehan C, Zhao Q, et al. QFMatch: multidimensional flow and mass cytometry samples alignment. Scientific Reports. 2018;8:3291.

[205] Maecker HT, Trotter J. Flow cytometry controls, instrument setup, and the determination of positivity. Cytometry Part A. 2006;69(9):1037-42.

[206] Richards AJ, Staats J, Enzor J, McKinnon K, Frelinger J, Denny TN, et al. Setting objective thresholds for rare event detection in flow cytometry. Journal of Immunological Methods. 2014;409:54-61.

[207] Costa E, Arroyo M, Pedreira C, Garcia-Marcos M, Tabernero M, Almeida J, et al. A new automated flow cytometry data analysis approach for the diagnostic screening of neoplastic B-cell disorders in peripheral blood samples with absolute lymphocytosis. Leukemia. 2006;20(7):1221-30.

[208] Pedreira CE, Costa ES, Lecrevisse Q, van Dongen JJ, Orfao A, Consortium E, et al. Overview of clinical flow cytometry data analysis: recent advances and future challenges. Trends in Biotechnology. 2013;31(7):415-25.

[209] Papoutsoglou G, Lagani V, Schmidt A, Tsirlis K, Cabrero DG, Tegnér J, et al. Challenges in the multivariate analysis of mass cytometry data: the effect of randomization. Cytometry Part A. 2019;95(11):1178-90.

[210] Bagwell CB, Adams EG. Fluorescence spectral overlap compensation for any number of flow cytometry parameters. Annals of the New York Academy of Sciences. 1993;677(1):167-84.

[211] Johnsson K, Wallin J, Fontes M. BayesFlow: latent modeling of flow cytometry cell populations. BMC Bioinformatics. 2016;17:25.

[212] Steen HB. Noise, sensitivity, and resolution of flow cytometers. Cytometry. 1992;13(8):822-30.

[213] Wang L, Hoffman RA. Standardization, calibration, and control in flow cytometry. Current Protocols in Cytometry. 2017;79(1):1-3.

[214] Giesecke C, Feher K, von Volkmann K, Kirsch J, Radbruch A, Kaiser T. Determination of background, signal-to-noise, and dynamic range of a flow cytometer: a novel practical method for instrument characterization and standardization. Cytometry Part A. 2017;91(11):1104-14.

[215] R Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria; 2013.

[216] Parks DR, Roederer M, Moore WA. A new "Logicle" display method avoids deceptive effects of logarithmic scaling for low signals and compensated data. Cytometry Part A. 2006;69(6):541-51.

[217] International Organization for Standardization. ISO 13528:2015 Statistical methods for use in proficiency testing by interlaboratory comparison. Geneva, Switzerland: ISO; 2015.

[218] Lee Rodgers J, Nicewander WA. Thirteen ways to look at the correlation coefficient. The American Statistician. 1988;42(1):59-66.

[219] Barnett D, Granger V, Storie I, Peel J, Pollitt R, Smart T, et al. Quality assessment of CD34+ stem cell enumeration: experience of the United Kingdom National External Quality Assessment Scheme (UK NEQAS) using a unique stable whole blood preparation. British Journal of Haematology. 1998;102(2):553-65.

[220] Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. Journal of Big Data. 2019;6(1):1-48.

[221] Lippeveld M, Knill C, Ladlow E, Fuller A, Michaelis LJ, Saeys Y, et al. Classification of human white blood cells using machine learning for stain-free imaging flow cytometry. Cytometry Part A. 2020;97(3):308-19.

[222] Yi X, Walia E, Babyn P. Generative adversarial network in medical imaging: A review. Medical Image Analysis. 2019;58:101552.

[223] Conrad VK, Dubay CJ, Malek M, Brinkman RR, Koguchi Y, Redmond WL. Implementation and validation of an automated flow cytometry analysis pipeline for human immune profiling. Cytometry Part A. 2019;95(2):183-91.