NAR Genomics and Bioinformatics, 2025, **7**, lqaf015
https://doi.org/10.1093/nargab/lqaf015
High Throughput Sequencing Methods

Polytect: an automatic clustering and labeling method for multicolor digital PCR data

Yao Chen [1,2,3,*], Ward De Spiegelaere [1,3,4], Wim Trypsteen [1,3,4,5,6], Jo Vandesompele [1,4,5,6,7], Gertjan Wils[7], David Gleerup[1,3,4], Antoon Lievens[1], Olivier Thas[1,2,8,9,*,†], Matthijs Vynck [1,3,†]

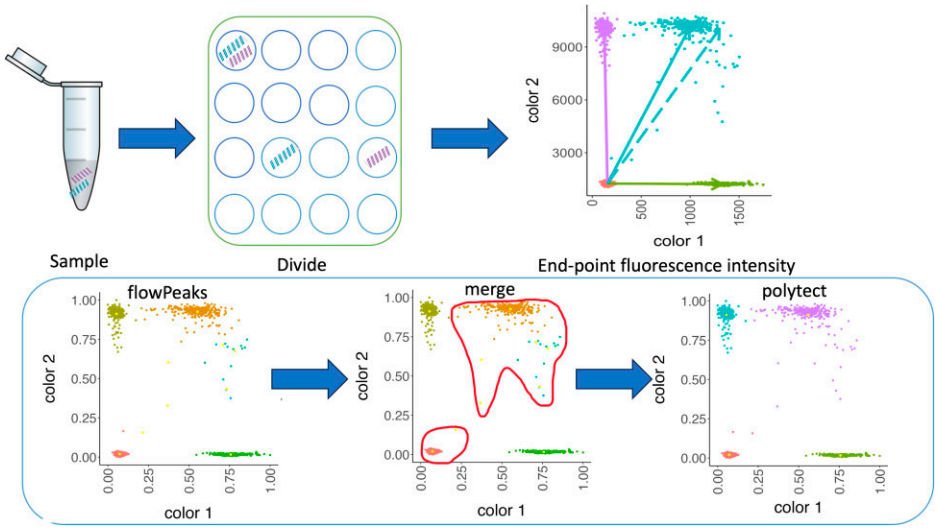
[1]Digital PCR Center (DIGPCR), Ghent University, 9820 Merelbeke, Belgium
[2]Department of Mathematics, Computer Science and Statistics, Ghent University, 9000 Ghent, Belgium
[3]Department of Morphology, Medical Imaging, Orthopaedics, Physiotherapy and Nutrition, Ghent University, 9820 Merelbeke, Belgium
[4]Cancer Research Institute Ghent (CRIG), Ghent University, 9000 Ghent, Belgium
[5]Department of Biomolecular Medicine, Ghent University, 9000 Ghent, Belgium
[6]OncoRNALab, Ghent University, 9000 Ghent, Belgium
[7]pxlence, 9000 Ghent, Belgium
[8]I-BioStat, Data Science Institute, Hasselt University, 3590 Hasselt, Belgium
[9]National Institute for Applied Statistics Research Australia (NIASRA), University of Wollongong, NSW 2522, Australia

[*]To whom correspondence should be addressed. Email: yao.chen@ugent.be
Correspondence may also be addressed to Olivier Thas. Email: olivier.thas@uhasselt.be
[†]The last two authors should be regarded as Joint Last Authors.

Abstract

Digital polymerase chain reaction (dPCR) is a state-of-the-art targeted quantification method of nucleic acids. The technology is based on massive partitioning of a reaction mixture into individual PCR reactions. The resulting partition-level end-point fluorescence intensities are used to classify partitions as positive or negative, i.e. containing or not containing the target nucleic acid(s). Many automatic dPCR partition classification methods have been proposed, but they are limited to the analysis of single- or dual-color dPCR data. While general-purpose or flow cytometry clustering methods can be directly applied to multicolor dPCR data, these methods do not exploit the approximate prior knowledge on cluster center locations available in dPCR data. We present Polytect, a method that relies on crude cluster results from flowPeaks, previously shown to offer good partition classification performance, and subsequently refines flowPeaks' results by automatic cluster merging and cluster labeling, exploiting the prior knowledge on cluster center locations. Comparative analyses with established methods such as flowPeaks, dpcp, and ddPCRclust reveal that Polytect often surpasses established methods, both on empirical and simulated data. Polytect manages to merge excess clusters, while also successfully identifying empty clusters when fewer than the maximally observable number of clusters are present. On par with recent developments in instruments, Polytect extends beyond two-color data. The method is available as an R package and R Shiny app (https://digpcr.shinyapps.io/Polytect/).

Graphical abstract



Received: September 20, 2024. **Revised:** February 8, 2025. **Editorial Decision:** February 12, 2025. **Accepted:** March 3, 2025

Introduction

Digital polymerase chain reaction (dPCR) is an increasingly popular molecular method that allows highly accurate, calibration-free quantification of nucleic acids [1]. Its advantages render it widely used across life science domains [2–5]. Unlike quantitative PCR, which requires real-time monitoring for quantification and a standard curve, dPCR typically relies on end-point fluorescence detection, i.e. fluorescence intensity at the end of the PCR amplification process, simplifying the reaction readout. Partition classification (thresholding; clustering) is based on these end-point fluorescence intensities. After thresholding, the binary or multinomial outcomes of the partitions allow for the quantification of the targeted nucleic acid(s) [5]. A dPCR glossary is available as Supplementary Table (Supplementary Table S1).

Recent advances in dPCR instrumentation give the user access to up to seven colors, enabling the simultaneous quantification of multiple target nucleic acids. However, multicolor dPCR data clustering poses challenges. Manual clustering is often performed, particularly for small series of single or duplex experiments. This may, however, introduce bias and low precision [6], and become increasingly difficult as the number of colors increases. We previously benchmarked automatic partition classification methods, including general-purpose, dPCR, and flow cytometry methods [7], and concluded that all methods face limitations, especially for the identification of small clusters or clusters with poor separation.

Some dPCR analysis methods make use of the expected cluster center locations [8, 9], whereas flow cytometry methods do not, as cluster positions are unpredictable in flow cytometry. Current dPCR partition classification methods primarily operate on two-color data, with some 1-color methods having the potential to be extended to the multicolor setting [10]. This contrasts with flow cytometry methods that could—in principle—be directly applied to multicolor, (higher order) multiplexed dPCR experiments.

To address these limitations, we developed "Polytect". The software is based on hierarchical mixture modeling and makes use of the expected dPCR cluster center locations. Unlike current dPCR methods, Polytect is applicable to multicolor, (higher order) multiplexed experiments. Polytect builds upon "flowPeaks" [11], recognized as the top-performing clustering algorithm in our previous benchmarking study [7]. Notably, flowPeaks demonstrated robust performance, even with very low target concentrations. However, a drawback of flowPeaks is that it occasionally yields more clusters than expected, posing challenges for automatic labeling and nucleic acid concentration estimation. In contrast to flow cytometry, the maximal number of clusters in a dPCR experiment is known, and their position is estimable. Polytect addresses this limitation by allowing users to specify the expected (maximum) number of clusters and by enabling automatic cluster labeling and nucleic acid concentration estimation. Polytect remains accurate even when some or all targets are absent.

We evaluated Polytect's performance using empirical and simulated data. The testing scenarios range from (higher order) two- to six-color data, detecting two to six targets in a reaction. Polytect's performance was compared with manual expert clustering and automatic methods flowPeaks, "dpcp", and "ddPCRclust".

Materials and methods

Methodology overview

The different stages of Polytect involve (i) preliminary flowPeaks clustering, (ii) merging of flowPeaks clusters via hierarchical modeling, (iii) integration of cluster position information through penalty terms, and (iv) automatic labeling and target concentration estimation. Fig. 1 and Algorithm 1 give an overview of the Polytect algorithm, and details are provided in the following sections.

Algorithm 1 Polytect

Require: Dataset $\boldsymbol{X}$, expected number of clusters k
Ensure: Cluster assignment, the estimated Gaussian mixture model $(\boldsymbol{\mu_h}, \boldsymbol{\Sigma_h}, \pi_h)$, $h = 1, ..., k$
1: Perform *flowPeaks* on $\boldsymbol{X}$
Initialization:
2: As initial center of the negative population $\boldsymbol{\mu_1}$, use $\hat{\boldsymbol{\mu}}_{\boldsymbol{1}}$ as estimated by *flowPeaks*, that is closest to (min(color1), min(color2), ...).
3: Initialize the centers of single positives as the 3rd quartile of intensities in each dimension
4: Initialize π_h as $1/k$
5: Initialize the $\boldsymbol{\Sigma_h}$ as the covariance of cluster centers given by *flowPeaks*. If the cluster contains one data point, then initialize using a very small number (e.g. 0.001).
Repeat:
6: **E step:** Estimate q_{gh} as in Eq. 6
7: **M step:** Estimate π_h, $\boldsymbol{\mu_h}$ and $\boldsymbol{\Sigma_h}$ as in Eqs. S7, S8 and S9, respectively.
Convergence

Initialization

In a preliminary step, flowPeaks is performed. The number of estimated clusters by flowPeaks can be, and often is, higher than expected. In the subsequent hierarchical mixture modeling step [12], excess clusters are merged. However, when fewer than the expected number of clusters are identified by flowPeaks, Polytect will fail to find the correct clusters as Polytect relies on cluster merging and is unable to split clusters. To address this issue, flowPeaks' parameters are tuned to obtain an excess number of clusters. This is achieved using Bayesian optimization with the "mlrMBO" R package (version 1.1.5.1, [13]). We specify the loss function as the discrepancy between the expected and observed number of clusters resulting from adjusting the input parameters for flowPeaks. As our method performs merging, we impose adjustments only when the actual number of clusters falls below the expected value. Of note, this adjustment is only performed during initialization but does not affect the subsequent cluster merging.

When there is only one cluster, only flowPeaks is performed: the subsequent steps are not executed, as Polytect performs merging only, and a single cluster cannot be merged further.

Hierarchical Gaussian mixture model

Suppose that the levels represent the steps required for clustering. Level $l + 1$ precedes level l. At level $l + 1$, more clusters are identified than expected, while at level l, these clusters have been further merged.

Assume there are k true clusters. At level $l + 1$ where only flowPeaks is performed, there are k_1 clusters. At the level l

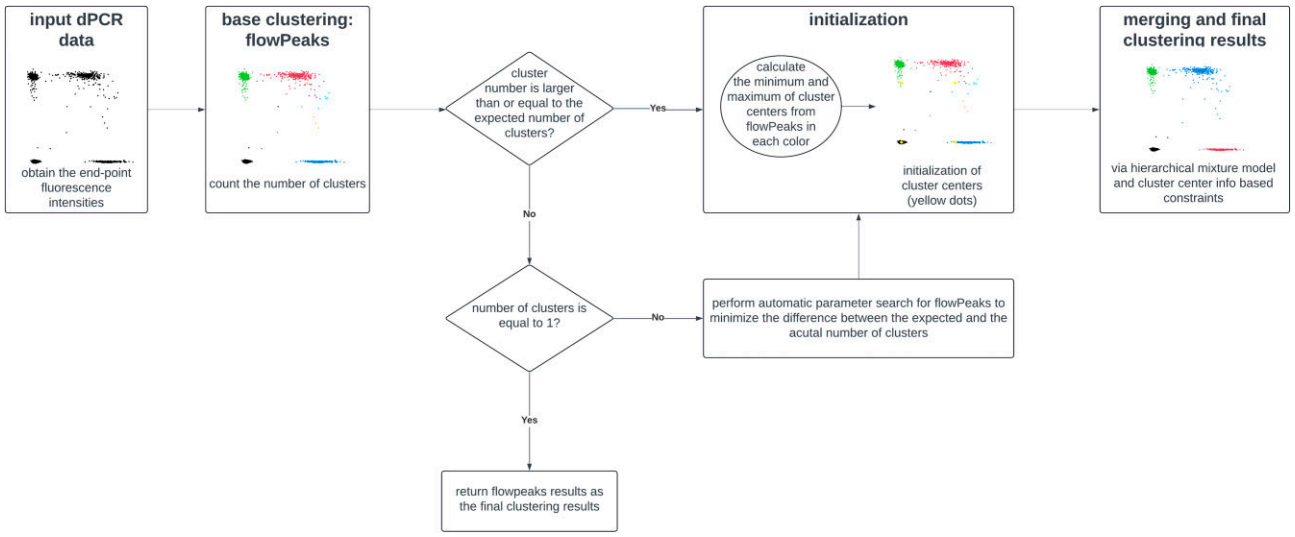


Figure 1. Schema of Polytect tool: (i) perform flowPeaks; (ii) initialize the cluster centers; and (iii) merge the cluster centers.

where the merging is performed, there remain k clusters. k_1 is often larger than k.

The likelihood of the observed data belonging to k_1 clusters at level $l+1$ can be written as [12]

$$L(X|\theta_{l+1}) = \prod_{i=1}^{n}\left[\sum_{g=1}^{k_1}\pi_g^{l+1} f(x_i|\theta_{l+1})\right], \quad (1)$$

where x_i represents the intensities of the ith partition (of a total of n partitions), π_g^{l+1} is the mixture weight (the fraction of the total partitions that is estimated to belong to the gth cluster) of component (cluster) g at level $l+1$, and $f(x_i|\theta_{l+1})$ is the probability density function of the ith data point x_i given the parameter set θ_{l+1}.

Data points assigned to a given cluster at level $l+1$ are assumed to be within the same cluster at level l as the method performs merging. The likelihood of the observed data at level l can be expressed as

$$L(X|\theta_l) = \prod_{g=1}^{k_1}\left[\sum_{h=1}^{k}\pi_h^l f(X_g|\theta_l)\right], \quad (2)$$

where X_g represents all the data points in the gth cluster.

Let $\boldsymbol{Z}$ denote the membership of clusters at level $l+1$ to clusters at level l. $\boldsymbol{Z}$ maps clusters across levels and is used in an Expectation–Maximization (EM) algorithm to estimate parameters. Specifically, z_{gh} is 1 or 0, indicating whether the gth component at level $l+1$ belongs to hth component at level l. The likelihood of the complete data at level l (given $\boldsymbol{Z}$) can be formulated as

$$L(X, Z|\theta_l) = f(X|Z, \theta_l) f(Z) \quad (3)$$

$$= \prod_{g=1}^{k_1}\prod_{h=1}^{k}\left[\pi_h^l f(X_g|z_{gh}, \theta_l)\right]^{z_{gh}} \quad (4)$$

and the log-likelihood becomes

$$l(X, Z|\theta_l) = \sum_{g=1}^{k_1}\sum_{h=1}^{k} z_{gh}\log(\pi_h^l f(X_g|z_{gh}, \theta_l)), \quad (5)$$

where X_g is the data belonging to component g at level $l+1$, π_h^l is the mixture weight of component h at level l, and $f(X_g|z_{gh}, \theta_l)$ is the probability density function of X_g given that it belongs to h at level l, parameterized by θ_l.

This log-likelihood function is used in an EM algorithm for estimating the parameters θ_l of the hierarchical mixture model at level l.

Parameter estimation via an EM algorithm

Here we assume a Gaussian mixture model at both level l and $l+1$. An EM algorithm is used to estimate the parameter set θ_l.

In the E-step, q_{gh}, the expected value of $z_{gh}|X_g$, θ_l can be derived as

$$q_{gh} = E[z_{gh}|X_g, \theta_l] = P(z_{gh} = 1|X_g, \theta_l) \quad (6)$$

$$= \frac{\left[G(\boldsymbol{\mu}_g^{l+1}, \boldsymbol{\mu}_h^l, \boldsymbol{\Sigma}_h^l) e^{-\frac{1}{2}tr(\boldsymbol{\Sigma}_h^{l\,-1}\boldsymbol{\Sigma}_g^{l+1})}\right]^{M_g}\pi_h^l}{\sum_k\left[G(\boldsymbol{\mu}_g^{l+1}, \boldsymbol{\mu}_k^l, \boldsymbol{\Sigma}_k^l) e^{-\frac{1}{2}tr(\boldsymbol{\Sigma}_k^{l\,-1}\boldsymbol{\Sigma}_g^{l+1})}\right]^{M_g}\pi_k^l}, \quad (7)$$

where $G(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the Gaussian density function with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. $\boldsymbol{\mu}_g^{l+1}$ is the mean of the gth component at level $l+1$, $\boldsymbol{\mu}_h^l$ is the mean of the hth component at level l, $\boldsymbol{\Sigma}_g^{l+1}$ is the covariance of the gth component at level $l+1$, and $\boldsymbol{\Sigma}_h^l$ is the covariance of the hth component at level l.

The M-step consists of maximizing the complete-data likelihood with regard to θ_l, resulting in

$$Q = \sum_{g=1}^{k1}\sum_{h=1}^{k} q_{gh} log(\pi_h^l f(X_g|z_{gh} = 1, \theta_l)). \quad (8)$$

More details can be found in the "EM algorithm" section of the Supplementary material.

Penalization

To enforce constraints on cluster centers, penalty terms are introduced that penalize deviations from the expected cluster positions.

In the case of a common, noncompeting two-target, two-color assay design, we expect to observe (i) a negative cluster with low fluorescence intensities in both colors, (ii) a sin-

gle positive cluster representing partitions positive for target 1 only, aligning horizontally with the negative cluster, (iii) a single positive cluster representing partitions positive for target 2, aligning vertically with the negative cluster, and (iv) a double positive cluster representing partitions positive for both targets, with its center approximately the sum of the centers of the two single positives that is positive for a single target (Fig. 2). The double positive cluster has the same endpoint fluorescence as the two single positive clusters. In the vector space, it is the sum of coordinates of the centers of the single positive clusters. However, deviations from these expected positions may occur, such as the double positive cluster (e.g. top right cluster, Fig. 2A) deviating from its expected intensities for color 1 (bottom right cluster, Fig. 2A). To accommodate such deviations, penalty terms are incorporated into Equation 8, resulting in

$$Q = \sum_{g=1}^{k1} \sum_{h=1}^{k} q_{gh} \log(\pi_h^l f(X_g|z_{gh} = 1, \theta_l)) - r_1 \|\boldsymbol{\mu_1} - \hat{\boldsymbol{\mu}}_1\|_2^2 - r_2 \|\boldsymbol{\mu_4} - \boldsymbol{\mu_2} - \boldsymbol{\mu_3} + \boldsymbol{\mu_1}\|_2^2, \quad (9)$$

where $\boldsymbol{\mu_1}, \boldsymbol{\mu_2}, \boldsymbol{\mu_3}, \boldsymbol{\mu_4}$ are the centers of the negative population, single positive (target 1), single positive (target 2), and double positive population, respectively. $\hat{\boldsymbol{\mu}}_1$ is the initial center estimate of the negative population. The user-defined parameter r_1 controls the deviation between the estimated initial cluster center of the negative population and the actual cluster center, and r_2 controls the deviation between the actual cluster center of the double positive population and the expected center. That is, $\boldsymbol{\mu_4} - \boldsymbol{\mu_1}$ should align as closely as possible with $\boldsymbol{\mu_2} - \boldsymbol{\mu_1} + \boldsymbol{\mu_3} - \boldsymbol{\mu_1}$. Details of the parameter estimation can be found in the "EM algorithm" section of the Supplementary material.

With these constraints, the cluster labels (i.e. the identification of what are the single and double positive clusters) are automatically determined. This method can be easily extended to (higher order) multiplexing settings [14] by imposing additional constraints. We provide mathematical formula derivations for higher order two- and four-color data in "EM algorithm" section of the Supplementary material.

Polytect can be extended to analyze other types of assays, such as competitive assays, by adding a constraint coefficient to the cluster centers. Equation 9 becomes

$$Q = \sum_{g=1}^{k1} \sum_{h=1}^{k} q_{gh} \log(\pi_h^l f(X_g|z_{gh} = 1, \theta_l)) - r_1 \|\boldsymbol{\mu_1} - \hat{\boldsymbol{\mu}}_1\|_2^2 - r_2 \|\boldsymbol{\mu_4} - \alpha_1 \boldsymbol{\mu_2} - \alpha_2 \boldsymbol{\mu_3} + (\alpha_1 + \alpha_2 - 1)\boldsymbol{\mu_1}\|_2^2, \quad (10)$$

where α_1 and α_2 are the coefficients used to construct a linear combination of the vectors formed by the cluster centers. They are chosen so that $\boldsymbol{\mu_4} - \boldsymbol{\mu_1}$ aligns as closely as possible with $\alpha_1(\boldsymbol{\mu_2} - \boldsymbol{\mu_1}) + \alpha_2(\boldsymbol{\mu_3} - \boldsymbol{\mu_1})$. α_1 and α_2 should be known and specified beforehand. In a noncompetitive assay, $\alpha_1 = \alpha_2 = 1$. For competitive assays, the fluorescence intensities of the double positives do not align with those of the single positives (see Supplementary Fig. S24 for an example). The typical fluorescence intensity of double positives in color 1 is ∼0.5 times the fluorescence intensity of single positive 1 in color 1. For color 2, it is ∼0.8 times the fluorescence intensity of single positive 2. In this case, we have $\alpha_1 = 0.5$, $\alpha_2 = 0.8$. For more details, please refer to "EM algorithm" section of the Supplementary material.

Performance evaluation

For the evaluation of the clustering methods, we selected four empirical two-color datasets (Fig. 2), three higher order two-color datasets, one three-color dataset, one four-color dataset, one five-color dataset, and two six-color datasets (Supplementary Tables S3–S8). These 12 datasets were obtained across 3 dPCR instruments.

For the two-color datasets, including the competitive assay and higher-order multiplexing datasets, we benchmarked Polytect against flowPeaks, dpcp, and ddPCRclust. For more than two-color datasets, the comparison was limited to flowPeaks, as dpcp and ddPCRclust are tailored to the two-color setting.

To ensure fair comparisons and mitigate biases stemming from differences in intensity scales across colors [7, 10], we conducted color-level min–max rescaling [15]. This procedure scales the data to a range [0, 1], preventing flowPeaks and dpcp from favoring colors with larger scales. Because ddPCRclust fails to cluster effectively when applied to such rescaled data, consistently yielding two clusters, we resorted to applying the method to the original, nonrescaled data.

When Polytect made significant errors on empirical data, such as failing to identify a cluster, we visually inspected the single-channel plots and increased the resolution, in order to identify the cause of the failure (Supplementary Tables S10 and S11).

Additionally, we tested the proposed method on simulated two-color data [7]. The simulation, based on empirical datasets, encompasses various scenarios on different factors (150 factor combinations in total), including low to high target concentrations, low to high percentages of rain, good to poor resolution, unimodal to bimodal distributions, orthogonal to non-orthogonal relationships, and equal to unequal target concentrations [7].

We used the adjusted rand index (ARI) [16], the relative bias of the estimated average number of target DNA molecules per partition (denoted as λ), and the quantities of interest (QOIs) to evaluate the clustering performance on both 12 empirical datasets and simulated data. The ARI quantifies the similarity between two cluster results of the same dataset. In particular, we compared the results of the proposed method with the expected labels of a reference clustering: true labels for simulated data and expert-determined clusters for empirical data. ARI scores range from −1 to 1, with 1 indicating perfect agreement, 0 indicating agreement not better than that obtained by random assignment of partitions to clusters, and −1 indicating complete disagreement. The relative bias of a λ estimate measures the deviation between the estimates and reference, relative to that reference, expressed as $(\hat{\lambda} - \lambda_{\text{ref}})/\lambda_{\text{ref}}$. Concerning the quantities of interest, most datasets were used for absolute quantification, except for the MM, CNV 5-plex, and CNV 6-plex datasets. In the case of absolute quantification, the impact on the quantity of interest is directly related to the relative bias of λ_s. The MM dataset was analyzed to measure DNA integrity, following the method described in [17]. The CNV 5-plex and CNV 6-plex datasets were analyzed to measure copy number variation. We reported the median of relative bias of λ_s and QOIs of the clustering methods across 100 simulations.

To match the clusters identified by flowPeaks to the reference populations, we employ the Hungarian assignment algorithm [18], which efficiently solves the linear assignment prob-

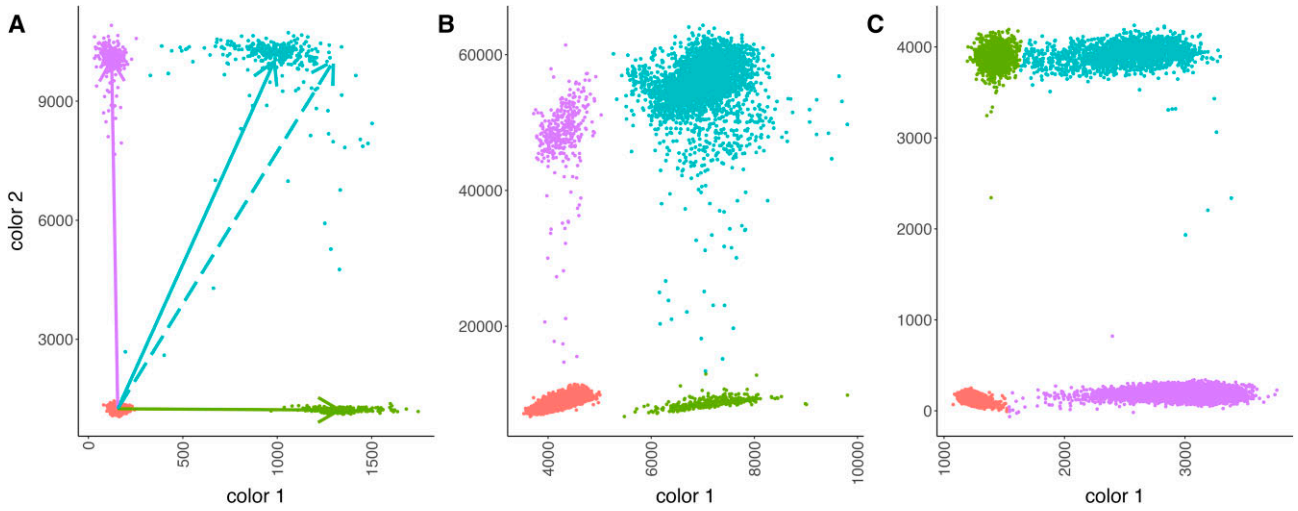


Figure 2. (**A**) HR (high-resolution) dataset. The solid arrow represents the actual cluster center, while the dashed arrow represents the expected cluster center. (**B**) MM (multi-mode) dataset. (**C**) LR (low-resolution) dataset.

lem by determining a one-to-one mapping that minimizes the total distance between the cluster centers provided by flowPeaks and those of the reference populations (the known true cluster centers). Polytect, dpcp, and ddPCRclust include automatic labeling.

We assessed the performance of the methods using optimal tuning parameter values for the empirical datasets [7]. Additionally, we recorded the run times for the empirical data analyses. To assess the stability of clustering results, we implemented the methods on 100 bootstrap samples drawn from the original datasets, each sample containing fluorescence intensities of 10 000 partitions. For the simulated data, we utilized the default parameter values, as conducting a thorough manual search for tuning parameters across all simulation scenarios was considered impractical. In the simulation setting, an automatic search may also be problematic [7]. However, we further examined scenarios where a method performed poorly, characterized by an ARI <0.8 or a relative bias $>20\%$ on both empirical and simulated data. For these cases, we optimized the methods' parameters (details are provided in "Parameter optimization" section of the Supplementary material, see Supplementary Figs S1–S5 and Supplementary Table S9).

Implementation, data, and code availability

We conducted all analyses using R (version 4.2.2) [19]. For the R package version, please refer to Supplementary Table S2 in "Package version" section of the Supplementary Data. In addition, we have developed an R package and a Shiny app that enable users to interactively explore different parameters and apply Polytect to a dataset of choice (see https://digpcr.shinyapps.io/Polytect/). R code and data are available from https://zenodo.org/records/14592424.

Results

Polytect achieved high ARI and low relative bias on empirical data

Across the 12 empirical datasets, Polytect consistently demonstrated strong performance, compared with other methods, emerging as the top performer in terms of ARI and relative bias (Table 1 and Supplementary Figs S11–S22). Polytect (median absolute relative biases: [0 to 6.70], sd: [0 to 2.28]) outperformed flowPeaks (median absolute relative biases: [0 to 46.44], sd: [0.052 to 19.31]), dpcp (median absolute relative biases: [0 to 13.63], sd: [0.064 to 11.86]) and ddPCRclust (median absolute relative bias: [0 to 8406.43], sd: [0.045 to 184]). This high relative bias of flowPeaks likely stems from cluster mislabeling, which is supported by the disconnection between the observed ARI (high) and the relative bias (high). Visual review indeed suggests appropriate clustering by flowPeaks (Supplementary Figs S3 and S12). However, the high dimensional setting renders automatic labeling difficult. For the five-color and six-color data, Polytect (median absolute relative biases: [0 to 0.46], sd: [0 to 1.18]) also did better than flowPeaks (median absolute relative biases: [0.09 to 1.08], sd: [0.35 to 2.76]).

Polytect often outperformed other methods in terms of accurately quantifying the quantity of interest (Table 1). In terms of relative bias for DNA integrity, Polytect achieved a significantly lower bias (0.15) compared with flowPeaks (2.41), dpcp (1.18), and ddPCRclust (14.64). Similarly, for the CNV 5-plex and CNV 6-plex datasets, Polytect demonstrated excellent performance with low relative biases for CNV (0 and 0.24, respectively) compared with flowPeaks (-2.96 and 0.24, respectively).

Performance of Polytect when compared with the automatic thresholding provided by dPCR instrument-specific proprietary software was mixed (Table 1 and Supplementary Figs S6–S10). For the HR, LR, CA, BPV, and CNV 5-plex datasets, the proprietary software produced results similar to those obtained by manual thresholding, with ARI values of 1 and relative biases close to 0. For these datasets, Polytect performed nearly as good (median ARI: [0.996 to 1]; median absolute relative biases: [0 to 1.04]). For the MM dataset, the proprietary software misclassified many data points at the edge of clusters, and rain (median ARI 0.988, Supplementary Fig. S7) while Polytect performed better, achieving a higher median ARI (0.997). For the higher order multiplexing datasets (HO-HIGH, HO-MED, and HO-LOW), the proprietary software could not be used for quantification because the number of targets exceeded the number of channels and the software only provided one threshold for each channel, making some clusters indistinguishable. Poly-

Table 1. Performance metrics from the resampling study of the empirical data

Dataset	Method	$\frac{\hat{\lambda}_1-\lambda_1}{\lambda_1}$ (%)	$\frac{\hat{\lambda}_2-\lambda_2}{\lambda_2}$ (%)	$\frac{\hat{\lambda}_3-\lambda_3}{\lambda_3}$ (%)	$\frac{\hat{\lambda}_4-\lambda_4}{\lambda_4}$ (%)	$\frac{\hat{\lambda}_5-\lambda_5}{\lambda_5}$ (%)	$\frac{\hat{\lambda}_6-\lambda_6}{\lambda_6}$ (%)	QOI (%)	ARI
HR	Polytect	−0.89	−0.29	/	/	/	/	/	0.999
	flowPeaks	−1.87	−0.60	/	/	/	/	/	1
	dpcp	−1.87	−0.62	/	/	/	/	/	0.999
	ddPCRclust	0.33	−0.30	/	/	/	/	/	0.999
	Biorad quantaSoft	−0.24	0.42	/	/	/	/	/	0.999
MM	Polytect	0.19	−0.39	/	/	/	/	0.15	0.997
	flowPeaks	2.07	−0.23	/	/	/	/	2.41	0.996
	dpcp	−0.01	−0.30	/	/	/	/	1.18	0.960
	ddPCRclust	17.04	0.24	/	/	/	/	14.64	0.973
	Stilla crystal Miner	2.6	0.43	/	/	/	/	−2.5	0.988
LR	Polytect	−1.04	−0.13	/	/	/	/	/	0.996
	flowPeaks	−1.18	−0.13	/	/	/	/	/	0.996
	dpcp	−3.35	0	/	/	/	/	/	0.985
	ddPCRclust	−0.06	0	/	/	/	/	/	0.998
	Biorad quantasoft	0	0	/	/	/	/	/	1
CA	Polytect	−3.77	−0.26	/	/	/	/	/	1
	flowPeaks	−20.33	−1.08	/	/	/	/	/	0.999
	dpcp	−10.55	−0.32	/	/	/	/	/	0.996
	ddPCRclust	8406.43	−2.47	/	/	/	/	/	0.865
	Roche digital LightCycler development	0	0	/	/	/	/	/	1
HO-HIGH	Polytect	0	−0.32	−0.23	/	/	/	/	0.999
	flowPeaks	0	−0.39	−0.23	/	/	/	/	0.998
	dpcp	−1.76	−0.72	13.63	/	/	/	/	0.994
	ddPCRclust	0.29	0.23	0.46	/	/	/	/	0.999
	Biorad quantasoft	/	/	/	/	/	/	/	0.999
HO-MED	Polytect	0	0	0	/	/	/	/	1
	flowPeaks	0	−0.32	0	/	/	/	/	1
	dpcp	1.37	0.33	1.84	/	/	/	/	0.999
	ddPCRclust	0.33	2.12	3.09	/	/	/	/	0.996
	Biorad quantasoft	/	/	/	/	/	/	/	/
HO-LOW	Polytect	−2.16	3.05	3.69	/	/	/	/	1
	flowPeaks	−2.17	0.91	0.28	/	/	/	/	0.997
	dpcp	0.29	4.64	4.64	/	/	/	/	1
	ddPCRclust	4.07	3.75	15.24	/	/	/	/	0.983
	Biorad quantasoft	/	/	/	/	/	/	/	/
BPV	Polytect	0	0	0	/	/	/	/	1
	flowPeaks	0	0	0	/	/	/	/	1
	Stilla crystal Miner	1.94	0.97	1.3	/	/	/	/	1
HIV 4-plex	Polytect	−6.70	−0.40	−3.40	−0.81	/	/	/	0.985
	flowPeaks	−4.94	8.17	−0.38	−46.44	/	/	/	0.985
	Biorad quantasoft	/	/	/	/	/	/	/	/
CNV 5-plex	Polytect	0	−0.06	0.11	−0.46	0	/	0	0.998
	flowPeaks	3.90	−0.09	−0.22	−0.47	1.06	/	−2.96	0.998
	Roche digital LightCycler development	0	−0.1	0	−0.1	0.02	/	−0.13	1
HIV 6-plex	Polytect	−0.11	0.29	0.01	0.06	−0.26	0.14	/	0.991
	flowPeaks	−0.78	−0.49	−0.68	−0.73	−1.08	−0.57	/	0.993
	Biorad quantasoft	0	0	0	0	0	0	/	1
CNV 6-plex	Polytect	−0.09	−0.26	−0.09	−0.09	−0.29	−0.25	0.24	0.992
	flowPeaks	−0.09	−0.26	−0.09	−0.09	−0.29	−0.25	0.24	0.991
	Roche digital LightCycler development	FAIL	−0.02	−0.01	0.07	−0.14	−0.14	FAIL	FAIL

Median relative bias of λ_1, λ_2, λ_3, λ_4, λ_5, and λ_6, QOI, and the ARI calculated for all resampled 10 000 data points. The result with the optimal tuning parameter value is shown. HR, MM, and LR are noncompeting two-color assay data, CA is competing mutation data, HO-HIGH, HO-MED, and HO-LOW are higher order two-color three-target data, BPV is three-color data, HIV 4-plex and 6-plex are four-color and six-color data, respectively, CNV 5-plex and CNV 6-plex are five-color and six-color data, respectively. "/" means "not applicable". "FAIL" means the software was not able to distinguish between positive and negative partitions. All partitions were marked as positive with this automatic thresholding method.

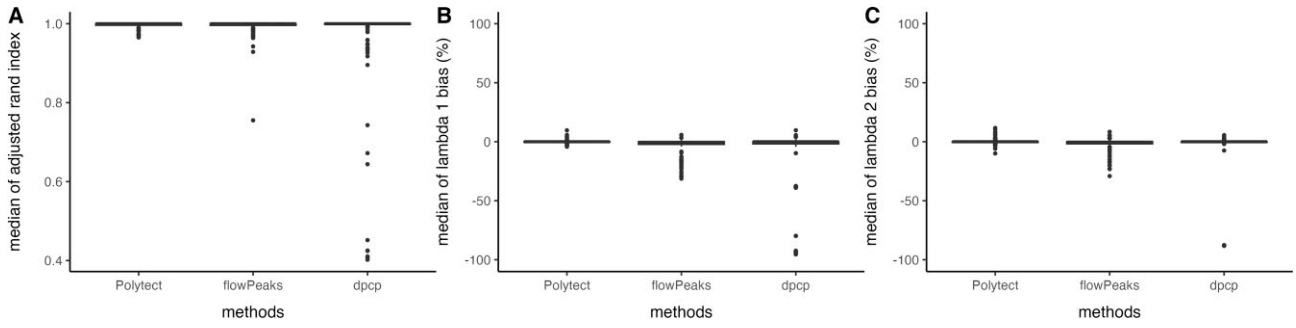


Figure 3. (**A**) Median ARI across the 150 factor combinations. (**B**) Median relative bias of λ_1 across the 150 factor combinations. (**C**) Median relative bias of λ_2 across the 150 factor combinations. The methods are ranked from best left panel to worst right panel. All methods perform well with median ARI close to 1 and bias of λ_s close to 0. Polytect outperforms flowPeaks and dpcp.

tect provided excellent clustering performance (median ARI [0.999 to 1]; median absolute relative biases [0 to 3.69]). For the CNV 6-plex dataset, the proprietary software failed to distinguish between positive and negative partitions in channel 1, mislabeling all partitions as positive. Contrarily, Polytect demonstrated robustness (median ARI 0.992; median absolute relative biases [0 to 0.29]).

Per sample run times were low overall ([0.61 to 16] s, see Supplementary Table S12), with Polytect requiring 1.14 s per sample, flowPeaks being fastest (0.61 s) and ddPCRclust the slowest (16.41 s). dPCP and ddPCRclust offer options to reduce runtime. After optimization, ddPCRclust was the fastest (0.21 s), while dPCP also showed improvement (from 3.84 to 3.06 s). For the details, please refer to section "Runtime" of the Supplementary Data.

Polytect outperformed other methods on simulated data

Across the simulation scenarios, all methods exhibited strong performance, with average median ARI values ranging from 0.954 to 0.996 and relative bias of λ_1 (absolute values) ranging from 0.13% to 10.6% (Fig. 3). Notably, Polytect demonstrated superior performance compared with both flowPeaks and dpcp in terms of ARI and relative bias. Moreover, Polytect exhibited smaller variation compared with flowPeaks and dpcp.

Polytect performed well in various scenarios

The effectiveness of Polytect was evaluated across various scenarios: (i) When more clusters than expected were identified by *flowPeaks* (Fig. 4A). (ii) When the expected number of clusters were identified by flowPeaks (Fig. 4B). (iii) When the actual number of clusters was 3, but the expected number was specified as 4 (Fig. 4C).

The results demonstrate that additional clusters were successfully merged by Polytect (Fig 4D). When the number of clusters identified by flowPeaks aligned with the expected number, Polytect refrained from merging, thereby retaining the cluster centers (Fig. 4E). In such cases, only automatic labeling was performed. However, when the expected number of clusters exceeded the actual count (e.g. four expected clusters for third actual ones), a cluster center was assigned without any data points belonging to this additional cluster.

In the case of the three-color BPV data, Polytect demonstrated effective performance, producing estimated cluster centers that closely matched those obtained by manual thresholding. Despite an expected presence of eight clusters for three targets, the triple positives are absent due to low target concentrations. Consequently, while Polytect provided a cluster center for the hypothetical triple positives, no partitions were assigned to this cluster (Table 2). When analyzing the four-color HIV data, Polytect effectively aligned cluster centers and sizes with the reference; however, it failed to identify a single positive cluster (+ − − −, see Supplementary Table S10 and Supplementary Figs S19 and S20. Supplementary Fig. S20 provides better visualization with clusters that are positive in other colors removed). This discrepancy may stem from a low resolution in color 1 (Supplementary Fig. S23). After we artificially increased the fluorescence intensity of the partitions above the threshold in color 1 by 0.2, the method successfully identified the missing cluster of 17 data points, aligning with the manual thresholding, and supporting that a low resolution caused Polytect to fail (Supplementary Table S11). In the case of the five-color and the six-color data, Polytect matches well with the manual thresholding results (Supplementary Fig. S25).

Methods failed on empirical and simulated data

When methods performed poorly, further manual parameter optimization often improved the results, but sometimes failed. For the CA dataset, both flowPeaks and ddPCRclust failed to provide accurate estimates of λ_1. While flowPeaks achieved high ARI by correctly classifying most partitions (Fig. 5A), it generated more clusters than expected (six rather than the expected four clusters), resulting in incorrect partition labeling. After parameter optimization, flowPeaks produced four clusters, reducing the absolute relative bias of λ_1 from 20.33% to 4.17% (Fig. 5D). ddPCRclust initially produced only two clusters and showed no improvement after parameter adjustments (Fig. 5E). For the HIV 4-plex dataset, flowPeaks struggled to identify many positive partitions in channel 4 (Fig. 5C). Initially, it generated only nine clusters (16 are expected). After optimization to match the expected number of clusters, flowPeaks produced 17 clusters; however, these were small, often comprising just one or two partitions. Due to limitations in the labeling method, the results did not improve significantly even after parameter optimization (Fig. 5F). In the simulated dataset, Polytect generally achieved high ARI and low bias. flowPeaks failed in five cases (3.3%), while dpcp failed in 25 cases (16.7%). To match the number of flowPeaks failure cases, for dpcp, we visually inspected the five worst cases (with the lowest ARI). Parameter tuning improved the results for both methods in these failure scenarios (Supplementary Figs S3 and S5).

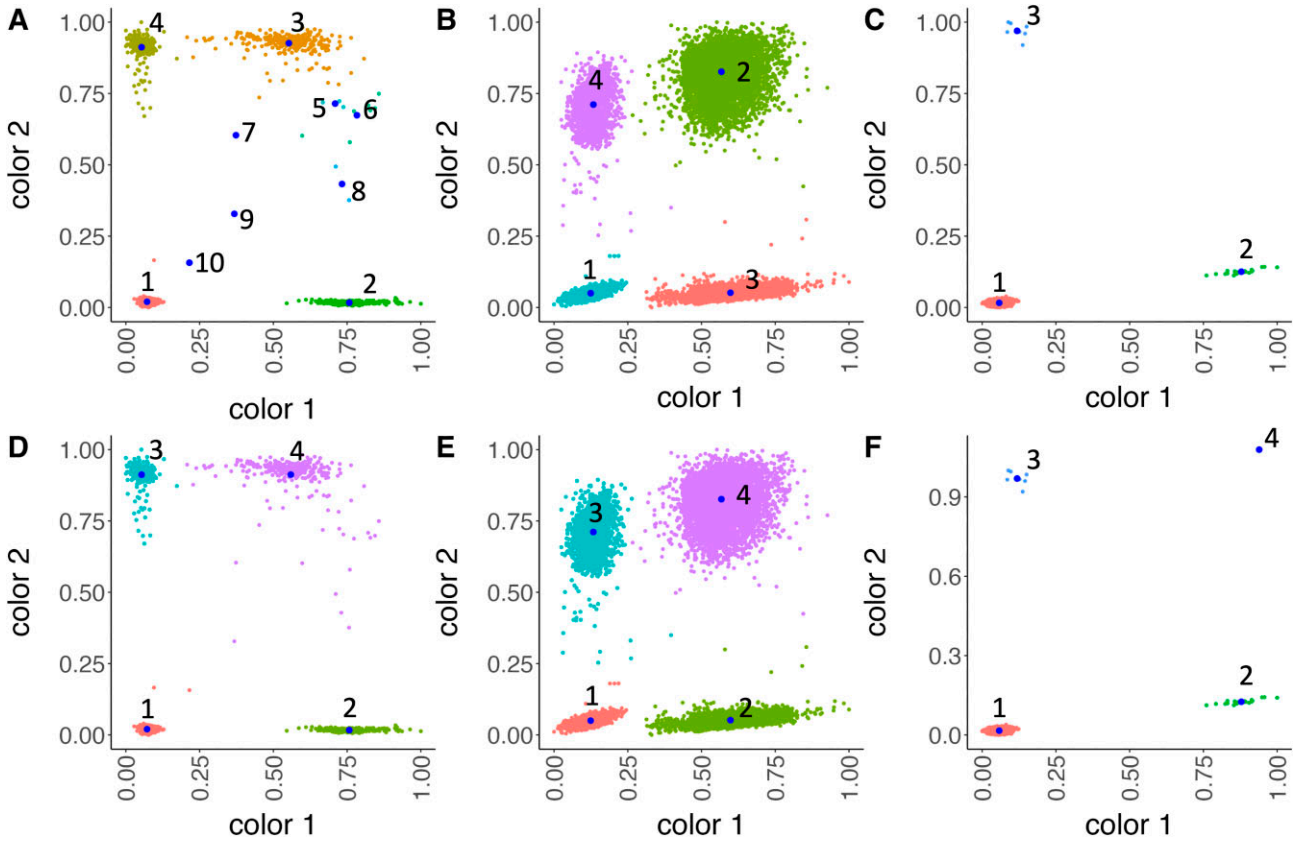


Figure 4. Clustering results obtained from flowPeaks and Polytect. The first row represents the clusters identified by flowPeaks, while the second row depicts those by Polytect. Each column corresponds to a different dataset: (**A**) and (**D**) HR dataset; (**B**) and (**E**) MM dataset; and (**C**) and (**F**) simulated data at very low concentration. The cluster centers are highlighted with a dot and a number. In first case, flowPeaks produced more clusters than expected (panel A), but Polytect successfully merged the surplus clusters (panel D). In second case, flowPeaks generated the expected number of clusters (panel B), and Polytect only relabeled the clusters in (panel E). In third case, flowPeaks produced fewer clusters than expected (panel C), and Polytect did not assign a partition to the double-positive cluster in (panel F).

Table 2. Cluster centers and sizes given by manual thresholding and Polytect on three-color BPV data

Cluster label	Methods	Cluster center	Cluster size
− − −	manual	(0.057, 0.086, 0.105)	24 401
	Polytect	(0.057, 0.086, 0.105)	24 401
+ − −	manual	(0.876, 0.074, 0.094)	481
	Polytect	(0.875, 0.074, 0.094)	482
− + −	manual	(0.056, 0.814, 0.087)	245
	Polytect	(0.056, 0.814, 0.087)	245
− − +	manual	(0.056, 0.079, 0.87)	323
	Polytect	(0.056, 0.079, 0.87)	323
+ + −	manual	(0.746, 0.558, 0.141)	4
	Polytect	(0.744, 0.618, 0.161)	3
+ − +	manual	(0.748, 0.084, 0.787)	6
	Polytect	(0.748, 0.084, 0.787)	6
− + +	manual	(0.057, 0.530, 0.733)	1
	Polytect	(0.057, 0.530, 0.733)	1
+ + +	manual	/	0
	Polytect	(0.874, 0.795, 0.845)	0

"−" indicates the absence of the targets and "+" indicates the presence.

Discussion

We have developed and validated Polytect, an automatic multicolor dPCR data classification and labeling method. Unlike previous partition classification methods that are limited to one- or two-color data, Polytect can be applied to experiments using any number of colors. Further strengths of Polytect are its automatic cluster labeling component, a step necessary for subsequent target nucleic acid concentration estimation, through exploitation of expected cluster positions, and exploitation of prior knowledge on the (maximum) number of distinct clusters. The latter is used for cluster merging of preliminary flowPeaks clusters. Importantly, Polytect accommodates scenarios where certain clusters may be absent due to low concentrations or absence of target(s). When there are only negative samples and no target is present, the method still functions effectively. In such a scenario, only flowPeaks is performed. One advantage of flowPeaks is that it does not require prespecification of the number of clusters.

An increasing need for methods like Polytect stems from the latest developments in dPCR instrument hardware. Instruments now allow analysis of up to seven colors, corresponding to 2^7 (128) observable clusters for simple one-color, one-target seven-plex assays. Data complexity increases further when using, e.g. multiple probes for (a subset of) targets ("higher order multiplexing"). Indeed, for such assays, the number of clusters is typically (much) higher. Unfortunately, current state-of-the-art dPCR clustering methods are restricted to at most two colors. While some of these methods could be extended to accommodate more colors, a curse of dimensionality issues arises [10]. Therefore, the development of improved methods capable of accurately clustering partitions and accommodating more than two colors is imperative. Polytect, being inher-

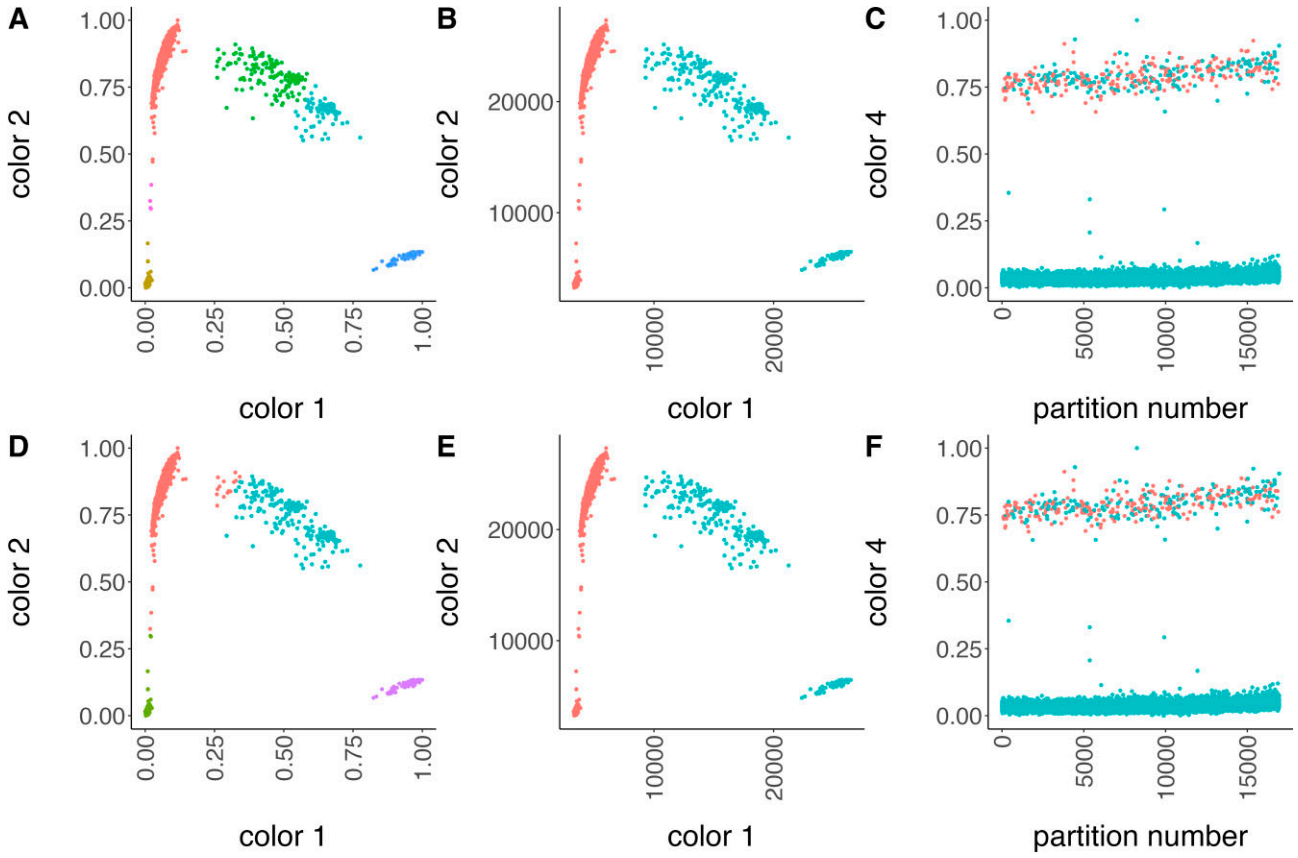


Figure 5. Scenarios where flowPeaks and ddPCRclust fail. The first row illustrates clusters identified before parameter optimization, while the second row shows clusters after optimization. Panels (A)–(D) correspond to the CA dataset, and panels (C) and (F) represent channel 4 of the HIV integrity-4 dataset. (**A**, **D**): Results from flowPeaks on the CA dataset. In panel (A), flowPeaks produced more clusters than expected. After parameter optimization, the surplus clusters were merged, as shown in panel (D). (**B**, **E**): Results from ddPCRclust on the CA dataset. In panel (B), ddPCRclust identified only two clusters, and this result did not improve even after optimization in panel (E). (**C**, **F**): Results from flowPeaks on channel 4 of the HIV integrity-4 dataset. In panel (C), flowPeaks failed to identify many positive partitions, misclassifying them as negative partitions. This issue persisted even after parameter tuning panel (F).

ently applicable to any number of colors, addresses this gap, and increases partition classification robustness through incorporation of prior knowledge on cluster location and count.

One example of Polytect's strengths is in rare mutation detection. In such experiments, one target (wild type) is often abundant while the other is rare or absent. Due to this low abundance, some single, double (triple, etc.) positive partition clusters will contain few or no partitions. Polytect exhibits high sensitivity in detecting such small clusters. Importantly, even in the absence of expected clusters, Polytect, will not split the largest partition cluster, unlike some generic clustering methods such as *kmeans*. Users benefit from the convenience of specifying only the expected (maximum) number of clusters, alleviating concerns regarding cluster absence. Additionally, Polytect's flexibility allows analysis of different applications, such as noncompetitive and competitive assays, the latter through fine-tuning the constraint coefficients of the cluster centers (Methods section).

We have conducted a comparative analysis of Polytect against dpcp, ddPCRclust, flowPeaks, and software provided by dPCR instruments. Across both empirical and simulated datasets, Polytect consistently outperformed or showed comparable performance to these methods. While flowPeaks, the base clustering algorithm for Polytect, demonstrates competence in handling multicolor data and detecting small clusters, its tendency to generate more clusters than expected poses challenges for automatic labeling. Because cluster locations in flow cytometry experiments are hard to predict, the cluster labeling issue arising in dPCR experiments is not addressed by flowPeaks. Polytect leverages the strengths of flowPeaks while addressing its limitation by merging excess clusters and automatically labeling clusters based on the prior knowledge of cluster center locations. Polytect's robustness on this front is supported by the excellent concordance of estimated cluster center locations with expertly assigned ones.

dpcp and ddPCRclust are constrained to (higher-order multiplexing) two-color datasets. Performance assessment shows that dpcp may misclassify (Fig. 3, outliers), potentially stemming from misidentification of negative or primary clusters. ddPCRclust yields inaccurate results when fluorescence intensity scales vary substantially across colors (Supplementary Fig. S14).

Polytect faces a few limitations. First, as it merges excess clusters using hierarchical Gaussian mixture modeling and relies on an EM algorithm for parameter estimation, the initialization of cluster centers is important. When initial cluster centers are too close together or too extreme, the algorithm may not converge. Second, the method labels automatically by imposing constraints on cluster centers, implying a requirement for position rules between the cluster centers

of single positive partition clusters, double positive partition clusters, and so forth. That is, we assume no interference between assays so that they are positioned in a near orthogonal way. However, if this assumption is violated, then an adaptation of the design is needed. As an example, we introduced a specific design for competing assays. For other assays, such as drop-off assays, further design changes would be needed. Moreover, since the method merges clusters rather than splitting them, it cannot generate more clusters than those produced by flowPeaks. In scenarios where data resolution is low (Supplementary Fig. S23), flowPeaks may fail to identify all clusters that actually exist, leading Polytect to overlook these clusters as well (Supplementary Table S10). While we try to alleviate this drawback by implementing the parameter optimization procedure outlined in [7], leading flowPeaks to identify more clusters, still fewer than the actual number is retrieved (9 versus 13 versus 14, before optimization, after optimization, and actual number, respectively). Utilizing different base clustering methods proficient in detecting small clusters is then a viable alternative. Indeed, Polytect is compatible with any initial clustering result.

In conclusion, our proposed automatic multicolor dPCR data clustering method, Polytect, can be applied beyond two-color data across different dPCR instruments and for various assay design types. It has demonstrated strong performance on both empirical and simulated data, achieving high ARIs and low relative biases. The method performs automatic labeling, making it convenient for analyzing data with three or more dimensions.

Acknowledgements

Authors contributions: Conceptualization: Y.C., M.V., and O.T.; Funding acquisition: O.T. and W.D.S.; Formal analysis: Y.C.; Supervision: M.V. and O.T.; Writing - original draft: Y.C.; Writing - review and editing: Y.C., W.D.S., W.T., J.V., D.G., A.L., O.T. and M.V.; Data curation: G.W. and W.T; Visualisation: Y.C., and Software: Y.C.

Supplementary data

Supplementary data is available at NAR Genomics & Bioinformatics online.

Conflict of interest

J.V. is co-founder of pxlence bv, providing universal Rainbow probes for digital PCR.

Funding

This work is funded in part by Bijzonder Onderzoeksfonds UGent (BOF, grant 01IO0420), and Agentschap voor Innoveren en Onderneme (VLAIO, grant HBC_2022.0673).

Data availability

R code and data are available from https://doi.org/10.5281/zenodo.14592424.

References

1. Huggett JF, Foy CA, Benes V *et al.* The digital MIQE guidelines: minimum information for publication of quantitative digital PCR experiments. *Clin Chem* 2013;**59**:892–902. https://doi.org/10.1373/clinchem.2013.206375
2. Querci M, Van den Bulcke M, Žel J *et al.* New approaches in GMO detection. *Anal Bioanal Chem* 2010;**396**:1991–2002. https://doi.org/10.1007/s00216-009-3237-3
3. Coccaro N, Tota G, Anelli L *et al.* Digital PCR: a reliable tool for analyzing and monitoring hematologic malignancies. *Int J Mol Sci* 2020;**21**:3141. https://doi.org/10.3390/ijms21093141
4. Tiwari A, Ahmed W, Oikarinen S *et al.* Application of digital PCR for public health-related water quality monitoring. *Sci Total Environ* 2022;**837**:155663. https://doi.org/10.1016/j.scitotenv.2022.155663
5. Hindson BJ, Ness KD, Masquelier DA *et al.* High-throughput droplet digital PCR system for absolute quantitation of DNA copy number. *Anal Chem* 2011;**83**:8604–10. https://doi.org/10.1021/ac202028g
6. Trypsteen W, Vynck M, De Neve J *et al.* ddpcRquant: threshold determination for single channel droplet digital PCR experiments. *Anal Bioanal Chem* 2015;**407**:5827–34. https://doi.org/10.1007/s00216-015-8773-4
7. Chen Y, De Spiegelaere W, Trypsteen W *et al.* Benchmarking digital PCR partition classification methods with empirical and simulated duplex data. *Brief Bioinform* 2024;**25**:bbae120. https://doi.org/10.1093/bib/bbae120
8. De Falco A, Olinger CM, Klink B *et al.* Digital PCR cluster predictor: a universal R-package and shiny app for the automated analysis of multiplex digital PCR data. *Bioinformatics* 2023;**39**:btad282. https://doi.org/10.1093/bioinformatics/btad282
9. Brink BG, Meskas J, Brinkman RR. ddPCRclust: an R package and Shiny app for automated analysis of multiplexed ddPCR data. *Bioinformatics* 2018;**34**:2687–9. https://doi.org/10.1093/bioinformatics/bty136
10. Vynck M, Chen Y, Gleerup D *et al.* Digital PCR partition classification. *Clin Chem* 2023;**69**:hvad063. https://doi.org/10.1093/clinchem/hvad063
11. Ge Y, Sealfon SC. flowPeaks: a fast unsupervised clustering for flow cytometry data via K-means and density peak finding. *Bioinformatics* 2012;**28**:2052–8. https://doi.org/10.1093/bioinformatics/bts300
12. Vasconcelos N, Lippman A. Learning mixture hierarchies. *Adv Neur Inf Proc Syst* 1998;**11**:606–12.
13. Bischl B, Richter J, Bossek J *et al.* mlrMBO: a modular framework for model-based optimization of expensive black-box functions. arXiv, https://doi.org/10.48550/arXiv.1703.03373, 3 December 2018, preprint: not peer reviewed.
14. Whale AS, Huggett JF, Tzonev S. Fundamentals of multiplexing with digital PCR. *Biomol Detect Quantif* 2016;**10**:15–23. https://doi.org/10.1016/j.bdq.2016.05.002
15. Patro S, Sahu KK. Normalization: a preprocessing stage. arXiv, https://arxiv.org/abs/1503.06462, 19 March 2015, preprint: not peer reviewed.
16. Hubert L, Arabie P. Comparing partitions. *J Classif* 1985;**2**:193–218. https://doi.org/10.1007/BF01908075
17. Gleerup D, Chen Y, Van Snippenberg W *et al.* Measuring DNA quality by digital PCR using probability calculations. *Anal Chim Acta* 2023;**1279**:341822. https://doi.org/10.1016/j.aca.2023.341822
18. Papadimitriou CH, Steiglitz K. Combinatorial Optimization: Algorithms and Complexity, North Chelmsford, MA: Courier Corporation, 1988.
19. R Core Team. R: a language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing, 2022.

Received: September 20, 2024. **Revised:** February 8, 2025. **Editorial Decision:** February 12, 2025. **Accepted:** March 3, 2025