mclUMI: Markov clustering of unique molecular identifiers enables dynamic removal of PCR duplicates

Jianfeng Sun[1,2,#] , Shuang Li[3], Stefan Canzar[4] & Adam P. Cribbs[1,2]

[1] Botnar Research Centre, Nuffield Department of Orthopaedics, Rheumatology and Musculoskeletal Sciences, National Institute of Health Research Oxford Biomedical Research Unit (BRU), University of Oxford, Oxford, UK.
[2] Oxford Centre for Translational Myeloma Research, University of Oxford, Oxford OX3 7LD, UK
[3] Gene Center, Ludwig-Maximilians University of Munich, Munich, Germany
[4] Faculty of Informatics and Data Science, University of Regensburg, 93053 Regensburg, Germany

Correspondence: jianfeng.sun@ndorms.ox.ac.uk

Abstract

Molecular quantification in high-throughput sequencing experiments relies on accurate identification and removal of polymerase chain reaction (PCR) duplicates. The use of Unique Molecular Identifiers (UMIs) in sequencing protocols has become a standard approach for distinguishing molecular identities. However, PCR artefacts and sequencing errors in UMIs present a significant challenge for effective UMI collapsing and accurate molecular counting. Current computational strategies for UMI collapsing often exhibit limited flexibility, providing invariable deduplicated counts that inadequately adapt to varying experimental conditions. To address these limitations, we developed mclUMI, a tool employing the Markov clustering algorithm to accurately identify original UMIs and eliminate PCR duplicates. Unlike conventional methods, mclUMI automates the detection of independent communities within UMI graphs by dynamically fine-tuning inflation and expansion parameters, enabling context-dependent merging of UMIs based on their connectivity patterns. Through *in silico* experiments, we demonstrate that mclUMI generates dynamically adaptable deduplication outcomes tailored to diverse experimental scenarios, particularly best-performing under high sequencing error rates. By integrating connectivity-driven clustering, mclUMI enhances the accuracy of molecular counting in noisy sequencing environments, addressing the rigidity of current UMI deduplication frameworks.

Keywords: Molecular quantification, PCR duplicate, UMI collapsing, Markov clustering, Gene expression, Genome analysis

Introduction

Genetic material is either fragmented for short-read sequencing or retained intact for long-read sequencing [1], laying the foundation for understanding genomic and transcriptomic

expression levels across various experimental conditions [2]. Accurate quantification of these fragments or entire molecules is a prerequisite for better dissecting their expression profiles [3,4]. However, directly quantifying original molecules from sequenced reads is challenging due to the presence of polymerase chain reaction (PCR) duplicates [5,6]. Therefore, the identification and removal of PCR duplicates are crucial for accurately estimating the number of unique molecules [7].

Unique molecular identifiers (UMIs) are sequences of oligonucleotides that are randomly synthesised to tag unique molecules, enabling accurate molecular counting [8–11]. The incorporation of UMIs has become a reliable method of choice due to its proven feasible across various high-throughput sequencing experiments and its effectiveness in demultiplexing during downstream analysis [12–14]. PCR duplicates with the same UMI are assumed to originate from the same molecule, a process known as UMI collapsing or grouping, which facilitates the counting of unique molecules. However, errors are frequently introduced into UMI regions of both original molecules and PCR duplicates. This results in significant numbers of unique molecules tagged with different UMIs, despite sharing the same origin, complicating the UMI collapsing process. To address this issue, different UMIs are collapsed together within an edit distance to tolerate errors and ensure accurate identification of true molecules. This error-tolerant strategy is effective in grouping different UMIs when reads undergo PCR amplification and sequencing at low error rates. In several sequencing technologies, such as single-cell RNA-seq (scRNA-seq), reads are subjected to numerous PCR amplification cycles to obtain sufficient genetic material for sequencing. However, research has shown that the accuracy of calculating gene expression profiles declines significantly at high PCR cycles because the number of PCR artefacts [15]. Consequently, the accuracy of molecular quantification varies in sequencing conditions, highlighting the need for computational strategies in UMI collapsing that are dynamically adaptable to specific experimental conditions.

In recent years, numerous computational tools for UMI collapsing have emerged [16–23]. These tools primarily focus on removing PCR duplicates by constructing graphs of UMIs [24,25]. One of the most widely used tools, UMI-tools, exemplifies graph-based strategies for effectively identifying unique UMI-tagged molecules. The process begins by constructing a graph where each node represents a unique UMI. An edge is drawn between the two nodes if their edit distance is no greater than a specified threshold, k; otherwise, they remain unconnected. This approach generates multiple connected components within the graph, where the number of components roughly corresponds to the unique UMI count. This is known as the *Cluster* method in UMI-tools. Within each connected component, further subdivision can occur. The number of subcomponents, where each central node (having the

highest count) is one edge away from the other nodes, is used to estimate the deduplicated molecule count. This approach is referred to as the *adjacency* method in UMI-tools. Since *cluster* and *adjacency* often led to an underestimated number and an overestimated number of unique UMIs, the *Directional* method was developed to seek for a more balanced estimation between somewhat of the two extremes. Its deduplication process can coarsely be described as a directed edge-visiting strategy, that is, merging node B by node A if the count of node A is at least two-fold greater than that of node B. The *directional* method in the *UMI-tools* suite is reported to gain the highest accuracy in identifying PCR duplicates [24]. However, regardless of the error settings or conditions, these strategies collapse UMIs into a single outcome once a set of reads is provided, potentially lacking the flexibility to dynamically expand or contract unique UMI counts to adapt to varying conditions.

Here, we present mclUMI, a graph clustering-based computational tool built using Markov clustering (MCL) algorithms [26–28], for precise localisation of unique UMIs and removal of PCR duplicates. Unlike established methods, the MCL-based approach identifies independent communities of UMIs within a connected component, each with its UMI nodes strongly connected and highly relevant. This process allows UMI nodes to be merged dynamically based on the connectivity of edges within a UMI graph. To incorporate the influence of UMI counts on the effectiveness of UMI collapsing, we developed two variant strategies, MCL-ed and MCL-val. Through comprehensive evaluations on simulated datasets with ground-truth information, we demonstrate that mclUMI achieves superior accuracy in molecular quantification compared to existing graph-based methods. As a complementary tool to state-of-the-art UMI identification frameworks, mclUMI enhances their applicability across heterogeneous experimental contexts.

Results

Markov clustering for UMI collapsing and PCR artefact removal

We developed mclUMI for automatic detection of UMI clusters by the Markov clustering algorithm without the need for calculating UMI counts, leading to the MCL method. It has two derivatives (MCL-ed and MCL-val) by considering the information about UMI counts (**Figure 1a**). As seen in **Figure 1b**, PCR artefacts or duplicates result from amplification of reads over multiple PCR cycles. Classification of PCR duplicates according to the same UMIs does not work when UMIs are synthesised and/or sequenced incorrectly because of DNA polymerase/sequencing errors. To tackle this problem, graph-based computational strategies have previously been proposed that leverage edit distances between UMIs to build UMI graphs (see [24] for technical review). Differently, mclUMI implements a Markov clustering algorithm for UMI collapsing based on a UMI graph. We began by simulating 7

UMIs for 10X V2 chemistry and constructing a UMI graph using an edit distance 1. Upon receiving the adjacency matrix of the UMI graph, mclUMI is asked to search for subcomponents with their respective nodes strongly connected (**Figure 1c**). For instance, a UMI graph composed of 7 nodes is further split into two subcomponents (**Figure 1d**). It shows high connectivity within each of them but low connectivity between them. Nevertheless, Markov clusters are discovered without leveraging the information on UMI counts. To include this information in MCL-based UMI collapsing, we additionally designed MCL-val, whose main thrust bears a very close resemblance to the Directional method. Rather than collapsing UMIs at the node level, MCL-val collapses UMIs at the subcomponent level. MCL-val identifies UMIs with the highest count from Markov clusters and merges two Markov clusters if the count of one of the most frequently observed UMIs is at least twofold higher than that of the other one. As illustrated in **Figure 1d**, because the difference in counts between UMIs A and B does not meet this requirement, the two Markov clusters remain separated**.** Building on the subcomponent level, we developed MCL-ed, which merges two Markov clusters if the two most frequently observed UMIs are within a k edit distance of each other. For the 7-node UMI graph, the two Markov clusters are merged because UMIs A and B are within one edit distance.

Comparison of UMI collapsing performance

We next evaluated the performance of UMI collapsing methods on simulated reads under varying conditions, including PCR cycle numbers, PCR errors, sequencing errors, UMI lengths, PCR amplification rates, and sequencing depths. In addition to the graph-based clustering methods in mclUMI, we implemented several Euclidean distance-based clustering methods, including DBSCAN, Birch, and Affinity propagation. Unlike mclUMI, which uses an adjacency adjacency matrix as input (**Figure 2a**), the distance-based clustering methods rely on non-graphical data for clustering UMIs. As illustrated in **Figure 2b**, UMI sequences are represented by one-hot encodings, which are then flattened and fed into the distance-based clustering algorithms. Our results demonstrate that methods implemented in mclUMI, along with DBSCAN, Birch, and *Directional* show a clear advantage in UMI collapsing over *Unique* and *Adjacency* across all six sequencing settings. Notably, mclUMI provides better UMI collapsing results than *Directional* at high PCR and sequencing error rates (**Figure 2c**). Additionally, we found that without building bespoke UMI graphs, DBSCAN and Birch can achieve a deduplication profile similar to UMI-tools methods. Consistent with the findings of the UMI-tools study [24], we observed that the number of unique molecules deduplicated by is underestimated at certain thresholds, particularly concerning varying UMI lengths and sequencing depths.

Additionally, we evaluated per-permutation UMI counts deduplicated by MCL-ed, MCL-val, and the UMI-tools *Directional* method across multiple sequencing error rates (**Supplementary Figure 2**). Our results show that both MCL-ed and MCL-val consistently yield lower fold changes in deduplicated UMI counts than the UMI-tools *Directional* method, highlighting that Markov clusters of UMIs can be post-processed and merged to further improve molecular quantification.

Markov clusters of UMIs efficiently capture UMIs from the same origin

The UMI-tools *directional* method excels in distinguishing original UMIs from amplified erroneous, amplified UMIs, making it a benchmark for UMI collapsing by leveraging information about both edit distance and UMI counts. To merge UMI nodes, the method initiates a series of directed visits from the UMI node with the highest count to those with lower counts. Ideally, when a node merges a neighbouring node into a cluster, reads attached with the two UMIs are derived from the same original molecule. To assess the effectiveness of the *directional* method in merging UMIs from the same origin, we labelled each amplified read the identifier of its original molecule during the read simulation process. This allows us to track the source of each read. As shown in **Figure 3a**, there are four possible scenarios, UMIs that are merged and derived from the same origin, UMIs that are merged but derived from a different origin, UMIs that are not merged but derived from the same origin, and UMIs that are not merged and derived from a different origin. The ideal outcome is to merge two UMIs that share the same root. Our results show that most of the methods effectively collapse UMIs from the same origin with regard to merging nodes (**Figure 3b**). Since only the MCL-val, MCL-ed, and *Directional* methods involve custom-built conditions for determining whether to merge UMIs, we further calculated the percentage and count of UMIs that are not merged from the same or different origin. Interestingly, there is a significant increase in the number of UMIs not merged by MCL-ed when they are from different origins, whereas the *Directional* method exhibits the opposite trend (**Figure 3c-e**). In summary, these findings suggests that merging UMIs nodes at the subcomponent level helps avoid the inclusion of reads from different original molecules compared to merging at the individual node level.

Determination of optimal parameters of Markov clustering algorithms for UMI collapsing

The MCL algorithm utilizes two parameters, *inflation* and *expansion*, to discover various clusters from a UMI graph. To determine the best choice of the parameters, we varied *inflation* in a range between 1 and 6 and *expansion* in a range between 2 and 9 and deduplicated PCR artefacts on simulated reads across 6 sequencing settings. We found that

both *inflation* and *expansion* induce the biggest perturbation on the deduplication effect when reads are subjected into a high PCR cycle number and large PCR/sequencing errors (**Supplementary Figure 3**). For dynamical clustering of UMIs, the *inflation* and *expansion* parameter are suggested to opt for a range between 1 to 3.5 and three integers 2, 3 and 4, respectively. Rather than an invariable result deduplicated by existing methods, this allows for generating various UMI clusters to achieve dynamic clustering.

Molecular quantification on simulated scRNA-seq data

We subsequently compared the accuracy of molecular quantification between mclUMI and UMI-tools at the single-cell sequencing level. We first used SPsimSeq [29] to generate a single-cell count matrix composed of 10 cells and 10 genes under various sequencing error rates and then simulated sequencing reads only for genes expressed in cells using ResimPy [15]. We found that in approximately 67.5% of cell-by-gene positions, mclUMI provided better UMI collapsing results than the UMI-tools *Directional* method at sequencing error rates ranging from 0.02 to 0.05 (**Figure 4**). Then, to obtain a count matrix with known cell types for visualisation purposes, we used DeepConvCVAE—trained on the PBMC68k dataset [30] and hosted in UMIche (https://2003100127.github.io/umiche)— to simulate a 1100×20736 cell-by-gene count matrix where 100 cells for each of 11 known cell types were sampled (**Figure 5a**). We found that the single-cell map for both mclUMI and *Directional* methods are in high agreement with the reference map classified by t-distributed stochastic neighbour embedding (TSNE) [31], particularly, at high sequencing error rates 0.05 and 0.1 (**Figure 5b-c**).

Case studies: connectivity of Markov clusters using mclUMI

Unlike existing methods, mclUMI emphasises the importance of identifying strongly connected subcomponents from a connected component. To illustrate how the connectivity between UMI nodes is determined, we analysed two connected components from simulated and iCLIP experiments [32], respectively. From simulated reads sequenced with an error rate of 0.05, we randomly selected a connected component with 67 unique UMIs a single edit distance apart. Using MCL, mclUMI gives two distinct subclusters, within which nodes are strongly connected but between which only two nodes are interconnected. By contrast, using the *Directional* method, UMI nodes are merged without considering the influence of the connectivity between UMIs, as indicated in the grey ellipse (**Figure 6a**). Similarly, **Figure 6b** illustrates that Markov clustering of 16 unique UMIs observed at a genomic position from the iCLIP dataset also captures strong connectivity between UMIs nodes. By contrast, the *Directional* method results in a few stray UMIs falling between or within clusters (e.g., the green UMI node). As UMIs in a cluster may be derived from the same UMI with the highest

count, we projected UMI counts onto UMI nodes to understand whether Markov clusters capture not only strong connectivity but also UMIs which are assumed to be representative of clusters. Our *in silico* results agree to this assumption that there are two UMIs that are overrepresented in two major clusters in red and sienna, respectively.

Furthermore, to assess how different UMI collapsing algorithms impact the identification of highly expressed genes, we plotted the expression profiles of the top 10 highly expressed genes using a single-cell dataset used in the original UMI-tools paper [24,33] (**Figure 6c**). We found that 6 out of 10 genes are identified identically by mclUMI and the *Directional* method.

Discussion

We developed mclUMI, a Markov clustering-based toolkit to efficiently remove PCR duplicates. The method leverages two key parameters—inflation and expansion—which control the detection of Markov clusters with strongly connected UMI relationships, offering greater flexibility and adaptability compared to established methods. An original molecule is often PCR-amplified multiple times, resulting in a high-count UMI. We hypothesise that representative UMIs with the high counts across different clusters may share the same origin. This is because an erroneous UMI, introduced through DNA polymerase errors in early PCR cycles, can be preferentially amplified in later cycles, leading to its overrepresentation. To address this, we implemented two post-processing strategies:

MCL-ed (edit distance-based merging): If representative UMIs from different Markov clusters are within a minimal edit distance (k), they are connected via edges and merged based on sequence similarity.

MCL-val (Count-Based Merging): If representative UMIs from different Markov clusters have a count difference within a t-fold threshold, they are linked and merged based on UMI abundance.

Analysis of original molecule identifiers across multiple PCR cycles confirms that both strategies effectively distinguish true molecules from PCR duplicates, improving the accuracy of molecular quantification.

One limitation of mclUMI is its higher computational cost compared to the cluster, adjacency, and directional methods, as it requires iterative optimisation to refine Markov clusters. Additionally, our exploration of clustering-based UMI collapsing has led to the development of Euclidean distance-based clustering algorithms, with DBSCAN and BIRCH demonstrating strong performance in UMI deduplication.

The impact of sequencing errors is particularly pronounced in long-read sequencing technologies, where error profiles are magnified. Our results show that mclUMI outperforms UMI-tools at high sequencing error rates, providing a valuable complement to existing computational methods for PCR artefact deduplication.

Methods

Reimplementation of UMI-tools deduplication methods

The *UMI-tools* methods are most widely used in UMI identification. To better understand and compare graph-based UMI collapsing algorithms, we reimplemented all deduplication methods reported in *UMI-tools*, including *cluster*, *adjacency*, and *directional*. The *cluster* and *adjacency* methods were constructed using the breadth first search (BFS) algorithm while the *directional* method was constructed using the depth first search (DFS) algorithm.

The reimplementation process begins by constructing a graph where each node represents a unique UMI. An edge is drawn between the two nodes if their edit distance is no greater than a specified threshold, k; otherwise, they remain unconnected. This approach generates multiple connected components within the graph, where the number of components roughly corresponds to the unique UMI count. This is known as the *Cluster* method in UMI-tools. Within each connected component, further subdivision can occur. The number of subcomponents, where each central node (having the highest count) is one edge away from the other nodes, is used to estimate the deduplicated molecular count. This approach is referred to as the *Adjacency* method in UMI-tools. Starting from node M with the highest UMI count in each connected component, a neighbour N connecting the node is grouped into the same connected subcomponent using the UMI-tools *Directional* strategy if C_M is at least twofold larger than C_N where C_M and C_N are the counts of nodes M and N. Next, after the first subcomponent in the connected component is formed, the node with the highest count from all remaining unvisited nodes is picked up for forming the second subcomponent. This process is repeated and aborted until all nodes in the connected component are visited only once, leaving K subcomponents.

Markov clustering of UMIs for collapsing

By calculating edit distances between UMIs and setting an editing distance cutoff, a graph is built to graphically present intricate relationships between UMIs. Edges that are used to link UMIs together reflect connectivity between them. If considering UMI counts as node sizes, such a graph can essentially reflect how UMIs with a low count evolve from a UMI with a high count over error-prone PCR amplification and sequencing processes. However, it remains murky as to which UMIs in a graph are most appropriately grouped and considered

to originate from the same origin. To cope with this challenge, UMI-tools meticulously designed a set of custom-built, expert-guided rules to construct two methods, *Adjacency* and *Directional*. Different from this notion, we conjecture that edges are highly informative to implicit relationships between UMIs and it might not suffice to find such relationships with custom-built rules especially from a large UMI graph containing a slew of edges. Building on this perspective, we set out to utilize Markov clustering (MCL) [34], a graph-based clustering algorithm, to naturally and spontaneously discover subcomponents with high connectivity from a UMI graph. The MCL-based UMI collapsing process is briefly formulated as follows.

We define V as a set of vertices $\{v_1, v_2, \ldots, v_n\}$ representing n unique UMIs from raw sequenced reads and E as a set of edges $\{e_1, e_2, \ldots, e_m\}$ $(E \subset V \times V)$ representing connections between UMIs. Specifically, UMI v_i can connect to UMI v_j as $e_{i,j}$ if v_i and v_j are k edit distance apart. Thus, with a given edit distance k we can obtain a graph $G_k = (V, E)$ to describe the relationships between UMIs. The adjacency matrix of G_k is denoted as A_k. Each element $t_{i,j}$ of the UMI transition matrix T_k of G_k is generated by applying a standardization operation to each element $a_{i,j}$ of A_k using the following equation.

$$t_{i,j} = \frac{a_{i,j}}{\sum_i^n a_{i,j}}$$

Subsequently, we generate a higher-order transition matrix D_k by taking the power l of T_k, expressed as

$$D_k = (T_k)^l$$

This step is termed the *expansion* of the initial UMI transition matrix T_k. Next, we impose an *inflation* operation on the element $d_{i,j}$ of D_k, such that

$$m_{i,j} = \frac{(d_{i,j})^r}{\sum_i^n (d_{i,j})^r}$$

where r is an integer. The *inflation* step [35] can enhance the detection of clusters with densely connected UMI nodes and remove weakly connected UMI nodes from clusters. The final transition probability $m_{i,j}$ in matrix M_k allows clustering of all the raw input UMIs. Based on massive *in silico* permutation tests, we empirically gained that the number of the obtained Markov clusters of UMIs might be large in some cases. For this reason, we further allowed the combination between Markov clusters at a UMI cluster level by introducing an edit distance k_M. Any two Markov clusters are combined if the centre UMIs (*i.e.*, with the highest UMI counts) of the two clusters share a less than k_M edit distance. In the mclUMI tool, this method is dubbed *mcl_ed*. We used *mclumi -m mcl_ed -gt XT -gist XS -ibam (input) -obam*

(output) for UMI deduplication throughout this research. Through trials, the expansion parameter r is set to 2 and the inflation parameter l ranges from 1.4 to 2.

For a bunch of reads, the biggest advantage of mclUMI over established methods (yielding only one UMI deduplication outcome) is the dynamic generation of different count matrices by tunning the l (expansion) and r (inflation). This property allows mclUMI to search fine-tuned parameters for better deduplication at different situations, e.g., high sequencing errors.

Post-processing strategies for merging Markov clusters of UMIs

Without considering UMI counts, UMIs are classified into different clusters with MCL based on UMI connectivity. Although the quantity of Markov clusters can be controlled within a certain range by *inflation* and *expansion* as introduced above, there might be association in distances or UMI counts between these clusters. Without considering UMI counts, erroneous UMIs that result from PCR or sequencing errors yet derive from the same ancestor UMI can be corrected by an edit distance to generate a cluster C_a with vertices $V_a = \{v_1^a, v_2^a, \ldots, v_n^a\}$. If considering UMI counts, within the cluster, UMI nodes are ideally supposed to evolve from a UMI node v_h^a with the highest count and PCR duplicates $N_{C_a}(v_h^a)$ are presumed to be all the neighbouring nodes of v_h^a, given by

$$N_{C_a}(v_h^a) = \{v_i^a \mid \{v_i^a, v_h^a\} \in C_a \ \vee i \neq h\}$$

We hypothesised that, at the inter-cluster level, two UMI nodes v_h^a and v_h^b with the highest counts from two clusters C_a and C_b (with vertices $V_b = \{v_1^b, v_2^b, \ldots, v_m^b\}$) in a small edit distance might be associated in terms of their origin. We further posit that a UMI could be incorrectly synthesised from the UMI of one of the original molecules at the early cycles of PCR amplification and thus gain a high chance to be chosen for amplification at the later cycles of PCR amplification, resulting in an overrepresentation of it in the final sequencing pool. In reality, however, the two UMIs share the same root. Given a set of representative UMI nodes $V_{a,b}^h = \{v_h^a, v_h^b\}$, there are $m + n - 1$ PCR duplicates, which are identified as the union of their neighbouring nodes $N_{C_{a,b}}(U_{a,b})$ and plus one of themselves. The former part is expressed as

$$N_{C_{a,b}}(U_{a,b}) = \bigcup_{u \in U_{a,b}} N_{C_{a,b}}(u)$$

where $C_{a,b}$ is the union of clusters C_a and C_b, which is built using the following edit distance.

$$dist^{ed}(v_h^a, v_h^b) = k$$

We refer to this edit distance-based strategy as MCL-ed to remove PCR deduplicates through merging of Markov clusters. This parameter is denoted as mcl_thres. Inspired by the UMI node merging strategy used in the *Directional* method, we further devise MCL-val, which merges UMI nodes at the Markov cluster (or subcomponents) level. Likewise, PCR duplicates are the neighbouring nodes $N_{C_{a,b}}(U_{a,b})$ that are considered to be derivatives of v_h^a or v_h^b if

$$c_{v_h^a} \geq 2C_{v_h^b} - 1$$

where $c_{v_h^a}$ is the count of v_h^a and $c_{v_h^b}$ is the count of v_h^b. As mentioned above, because two representative nodes can originate from one another at the very early PCR cycles, the following condition is also applicable to the merging of UMIs.

$$c_{v_h^b} \geq 2C_{v_h^a} - 1$$

Representation of UMI sequences

Let x be a nucleotide from set $S^{nt} = \{A, T, G, C\}$ and S be a UMI sequence of length d. Each nucleotide of S is one-hot encoded to be vector $\mathrm{X}^{nt} \in \mathbb{R}^{1\times 4}$

$$\mathrm{X}^{nt} = [x_1, x_2, x_3, x_4]$$

where x_j is given by:

$$x_j = \begin{cases} 1, & if\ j = i \\ 0, & if\ j \neq i \end{cases}$$

where i is the index of a nucleotide in S^{nt}.

Let X^S be the one-hot encoded matrix of S. After applied with a flatten operation, a UMI sequence is finally represented by $Y^S \in \mathbb{R}^{1\times 4d}$, such that

$$Y^S = flatten(\mathrm{X}^S)$$

which was fed into Euclidean distance-based clustering algorithms for clustering of UMI nodes that are highly relevant.

Euclidean distance-based clustering of UMIs

To remove PCR duplicates without building a UMI graph and further compare graph-based deduplication methods, we tested the performance of three Euclidean distance-based clustering algorithms for clustering of UMIs, including density-based spatial clustering of applications with noise (DBSCAN) [36], balanced iterative reducing and clustering using hierarchies (BIRCH) [37], and affinity propagation [38].

DBSCAN. Suppose we have t unique UMIs $\mathrm{S} = \{S_1, S_2, \dots, S_t\}$ observed at a given genomic position and their one-hot encodings are $\mathrm{Y} = \{Y^{S_1}, Y^{S_2}, \dots, Y^{S_t}\}$. Each UMI has d nucleotides. Let $MinPts$ be a predefined number of UMIs required for forming a dense region [39]. The Euclidean distance between any two UMIs is computed as follows

$$dist^{eu}(Y^{S_i}, Y^{S_j})$$

A UMI S_p is said to be a core point if there are at least

$$n_{S_p} \geq MinPts$$

where n_{S_p} is number of UMIs falling within a region D_ε where any UMI and S_p are within ε Euclidean distance, denoted as $N_{D_\varepsilon}(S_i)$

$$N_{D_\varepsilon}(S_i) = \{S_j \in D_\varepsilon \mid dist^{eu}(Y^{S_p}, Y^{S_q}) \leq \varepsilon\}$$

A UMI S_q is said to be a border point if the Euclidean distance between any core point and itself is no greater than ε but it has

$$n_{S_q} < MinPts$$

Last, any UMIs that are neither core nor border points are said to be noise points.

DBSCAN starts finding core points, iteratively assigns border points to the dense region of their associated core points, and stops running until all points are labelled a cluster.

BIRCH. As BIRCH pertains to complex machinery of clustering data [40], we provide a brief introduction as follow. BIRCH is representative of the hierarchical clusters methods, which utilizes a clustering feature (CF) to describe a cluster, represented as

$$CF \leftarrow (t, LS, SS)$$

where t represents the number of the unique UMIs as given above. LS refers to a linear sum operation on the one-hot vectors of UMIs, phrased as

$$LS = \{\sum_{i=1}^{t} Y^{S_i}(1), \sum_{i=1}^{t} Y^{S_i}(2), \dots, \sum_{i=1}^{t} Y^{S_i}(4d)\}$$

SS refers to a squared sum operation on the one-hot vectors of UMIs as follows

$$SS = \sum_{i=1}^{t} \sum_{j=1}^{4d} Y^{S_i}(j)^2$$

Then, BIRCH builds a CF tree by which CF hierarchical clusters are produced and their cluster centres and scales are updated by computing a few statistics including radii (measured using the Euclidean distance) and diameters.

Affinity propagation. Affinity Propagation identifies exemplars of clusters by leveraging three matrices [41], including a similarity matrix M^S, a responsibility matrix M^R, and an availability matrix M^A. Actually, it initially picks a few exemplars as cluster centroids of subsets of UMIs. We take one subset of UMIs $\mathrm{S}_s = \{S_1, S_2, \dots, S_c \mid c < t\} \subset \mathrm{S}$ as an example to illustrate how to determine an exemplar. We have $M^R \in \mathbb{R}^{c\times c}$, $M^A \in \mathbb{R}^{c\times c}$, and $M^A \in \mathbb{R}^{c\times c}$. $M^S = [dist^{eu}(Y^{S_i}, Y^{S_j})]_{c\times c}$ whose elements are the Euclidean distances between UMIs. Elements of M^R and M^A are updated using the following rules.

$$M^R_{i,j} \leftarrow M^S_{i,j} - \max_{k\neq j}(M^S_{i,k} + M^A_{i,k})$$

$$M^A_{i,j} \leftarrow \min(0, M^R_{j,j} + \sum_{k\neq i,j}^{c} \max\,(0, M^R_{k,j}))$$

Finally, a UMI at position i is chosen to be a cluster exemplar if it satisfies

$$\max_i(M^A_{i,j} + M^R_{i,j})$$

Evaluation metrics

The method performance for UMI identification is evaluated using the fold change (FC)

$$\frac{N_e - N_t}{N_t}$$

where N_e stands for the estimated number of UMIs and N_t stands for the true number of UMIs. It is used to measure a relative difference.

Alternatively, the performance comparison between methods is evaluated by the difference between fold changes dFC, which is written to be

$$\mathrm{dFC} = FC_{m1} - FC_{m2}$$

where $m1$ and $m2$ represent two deduplication methods and $FC_{m1} \geq 0$ and $FC_{m2} \geq 0$. $\mathrm{dFC} > 0$ indicates the performance of method $m1$ (or $m2$) better than method $m2$ (or $m1$).

Settings of simulated reads

We simulated sequencing reads at a single genomic locus to examine the performance of mclUMI. UMI lengths vary from 6 to 18. Reads are amplified from PCR cycles 1-16. Amplification rates are changed from 0.1-1. Sequencing error rates vary in a range [1e-05,

2.5e-05, 5e-05, 7.5e-05, 1e-04, 2.5e-04, 5e-04, 7.5e-04, 1e-03, 2.5e-03, 5e-03, 7.5e-03, 0.01, 0.025, 0.05, 0.075, 0.1, 0.2, 0.3]. Polymerase error rates during PCR amplification are set in a range [1e-05, 2.5e-05, 5e-05, 7.5e-05, 1e-04, 2.5e-04, 5e-04, 7.5e-04, 0.01, 0.05]. Sequencing depths vary in a range [100, 200, 500, 600, 800, 1000, 2000, 3000, 5000]. To make our experiment controllable, we generated sequencing data using one of the above parameter settings each time while keeping the rest of parameters fixed. Based on our empirical analysis and in line with the *UMI-tools* work, the fixed parameters of PCR cycles, PCR errors, sequencing errors, UMI lengths, and amplification rates were set to 8, 1e-05, 1e-03, 10, and 0.85, respectively. To eliminate random errors and biases, we carried out 10 permutation experiments over which the final fold change is averaged.

Formulation of read simulation

The total number of reads $n_r^{(i)}$ at PCR cycle i consists of two sources of reads, i.e., reads that are and are not PCR amplified. An initial number of reads is set to 50. The estimate of amplified reads follows a Galton-Watson branching (GWB) process [42,43], such that

$$n_r^{(i)} = \begin{cases} n_r^{(i-1)} + \omega^{(i)}, & if\ i > 0 \\ 50, & if\ i = 0 \end{cases}$$

where $\omega^{(i)}$ is the total number of reads resulting from PCR amplification and can be calculated from a binomial distribution

$$Binom(n_r^{(i-1)}, \mu)$$

where μ is the amplification efficiency.

Likewise, we use the same way to simulate PCR errors. The total number of errors $n_e^{(i-1)}$ at PCR cycle i is calculated as

$$n_e^{(i)} = \begin{cases} n_e^{(i-1)} + \tau^{(i)}, & if\ i > 0 \\ 0, & if\ i = 0 \end{cases}$$

where $\tau^{(i)}$ is the number of errors synthesised at PCR cycle i and is drawn from a negative binomial distribution [44]

$$NBinom((1-\epsilon) \times n_b^{(i)}, 1-\epsilon)$$

where ϵ is the PCR error rate and $n_b^{(i)}$ is the number of bases of amplified reads at PCR cycle i.

The number of sequencing errors n_e is estimated from a negative binomial distribution

$$NBinom((1-\vartheta) \times n_b, 1-\vartheta)$$

where ϑ is the sequencing rate and n_s is the total number of bases after PCR amplification.

Simulation of single-cell count matrices

We employed SPsimSeq [29] to generate an expression matrix of genes in single cells simulated based on the SingleCellExperiment data [45]. We simulated a single-cell count matrix with 10 genes and 10 cells to create an initial pool of 1828 unique molecules attached with cell barcodes and UMIs. The length of simulated UMIs is set to 10 in line with the protocol of10X Genomics V2 chemistry.

To supply information about cell types, we then used DeepConvCVAE trained by a variational autoencoder on the PBMC68k dataset [30] (https://github.com/10XGenomics/single-cell-3prime-paper/tree/master/pbmc68k_analysis) and generated a count matrix of 1100 cells (100 cells per each of 11 cell types) and 20,736 genes. The DeepConvCVAE framework and the trained model are currently housed in UMIche (https://2003100127.github.io/umiche).

Identification of highly expressed genes

We identified the top k highly expressed genes by

$$\overrightarrow{\overline{x_1^{j1}}, \overline{x_2^{j2}}, \overline{x_{k-2}^{j3}}, \ldots, \overline{x_k^{jk}}}$$

where $\overline{x_k}$ represents the identifier of a gene with the kth highest average gene expression value at column jk ($jk \in 1,2, \ldots, n$) of a UMI count matrix with m cells and n genes.

Code and data availability

The mclUMI software is publicly available at https://github.com/cribbslab/mclumi.

Declaration of competing interest

A.P.C is listed as an inventor on several patents filed by Oxford University Innovations concerning single-cell sequencing technologies. The other authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Author contribution

J.S. conceived this research. J.S. developed the algorithm, implemented mclUMI, and wrote the manuscript. J.S. conducted data analysis with contributions from S.L., S.C., and A.P.C. J.S. and A.P.C. acquired funding. All authors have given approval to the final version of the manuscript.

Acknowledgement

This work was financially supported by the Medical Research Council (MRC) career development fellowship (MR/V010182/1).

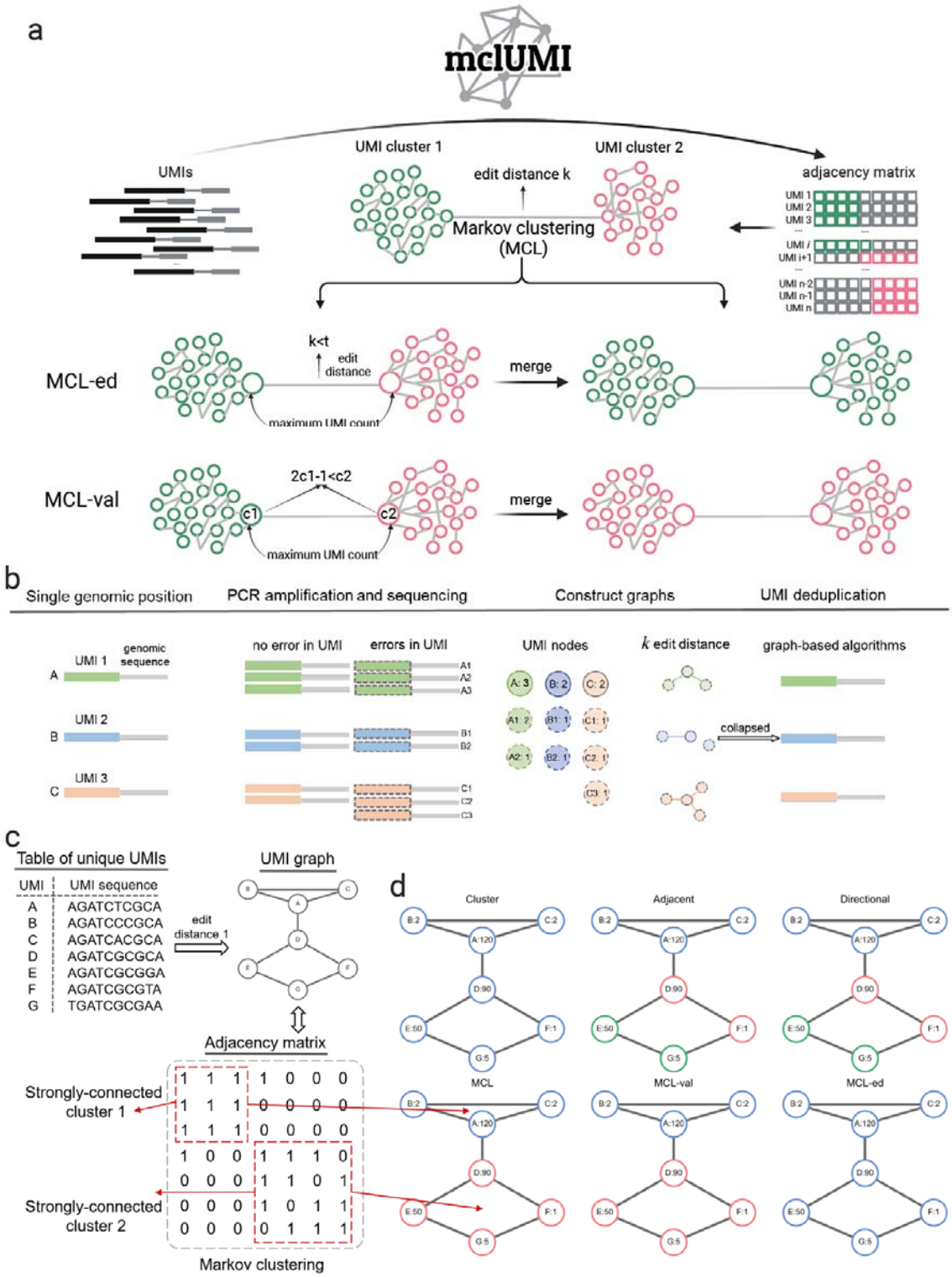


Figure 1. Overview of UMI deduplication using Markov clustering. **a**, Schematic of Markov clustering of UMIs in the mclUMI tool. **b**, Subjecting UMI-tagged molecules to PCR amplification and sequencing. The effectiveness of UMI deduplication is evaluated by how the estimated number of molecules is close to the actual number of molecules. **c**, Building a graph with 7 unique UMI nodes (A-G) connected with an edit-distance threshold of 1. Strongly connected components are visually displayed in the adjacency matrix. **d**, Graph-based identification of true molecules by collapsing UMIs using the cluster, adjacency, directional, mcl, mcl-ed, and mcl-val algorithms. PCR duplicates are merged into different UMI groups in different edge colours. The identifier of each unique UMI is followed by its count in the single genomic locus.

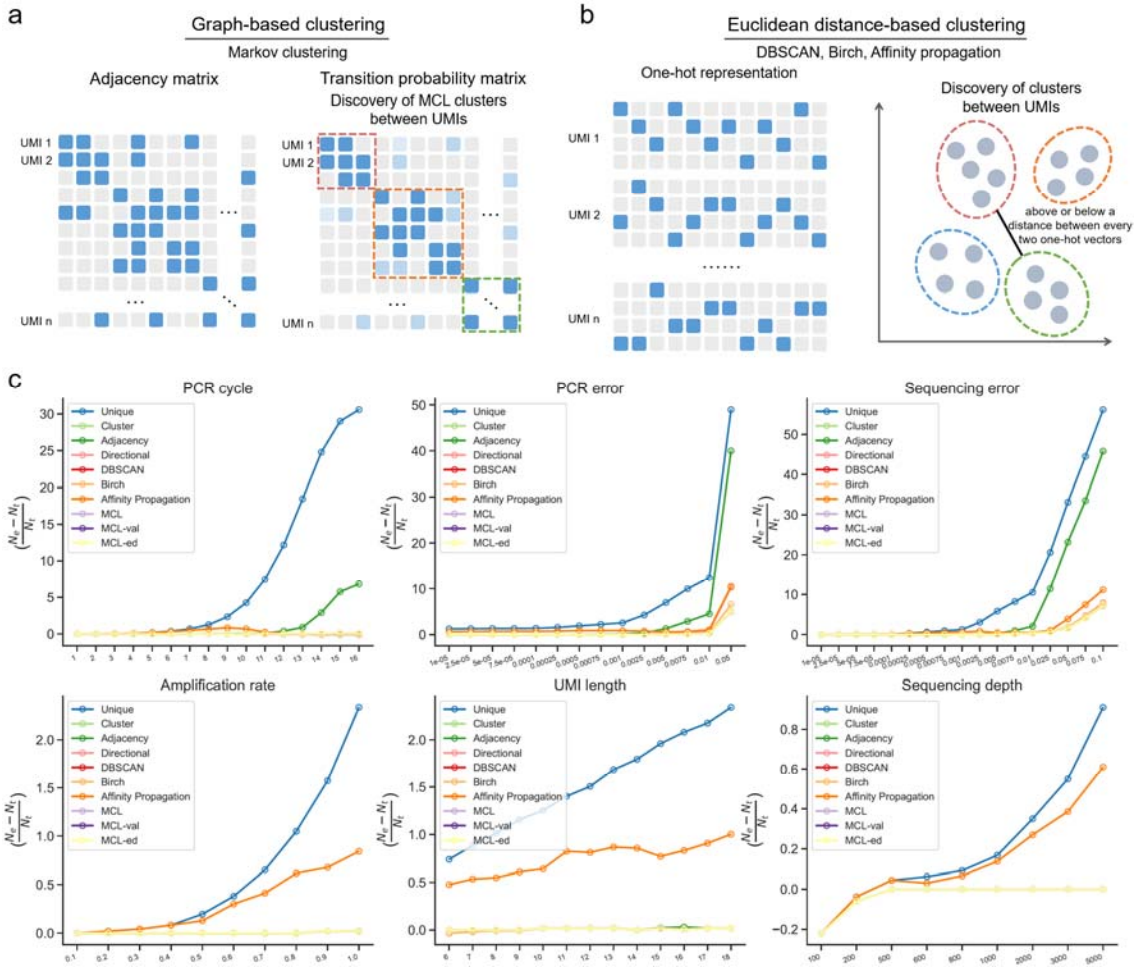


Figure 2. Performance comparison of UMI collapsing methods. **a**, Schematic of graph-based clustering methods for identifying UMI clusters based on the adjacency matrix and the transition probability matrix. **b**, Schematic of Euclidean distance-based clustering methods for identifying UMI clusters using one-hot representations of UMIs. One position in each

column is occupied to be indicative of its nucleotide type. The one-hot matrix is flattened as a vector with a dimension of 1x4n (*n* represents the UMI length) prior to taking them as input to the Euclidean distance-based clustering methods. **c**, Comparison of deduplication results between methods across multiple PCR cycles, PCR error rates, sequencing error rates, amplification rates, UMI lengths, and sequencing depths, respectively. A close-up view of the deduplicated counts of Directional, MCL, MCL-ed, MCL-val, DBSCAN, and Birch can be found in **Supplementary Fig. 1**.

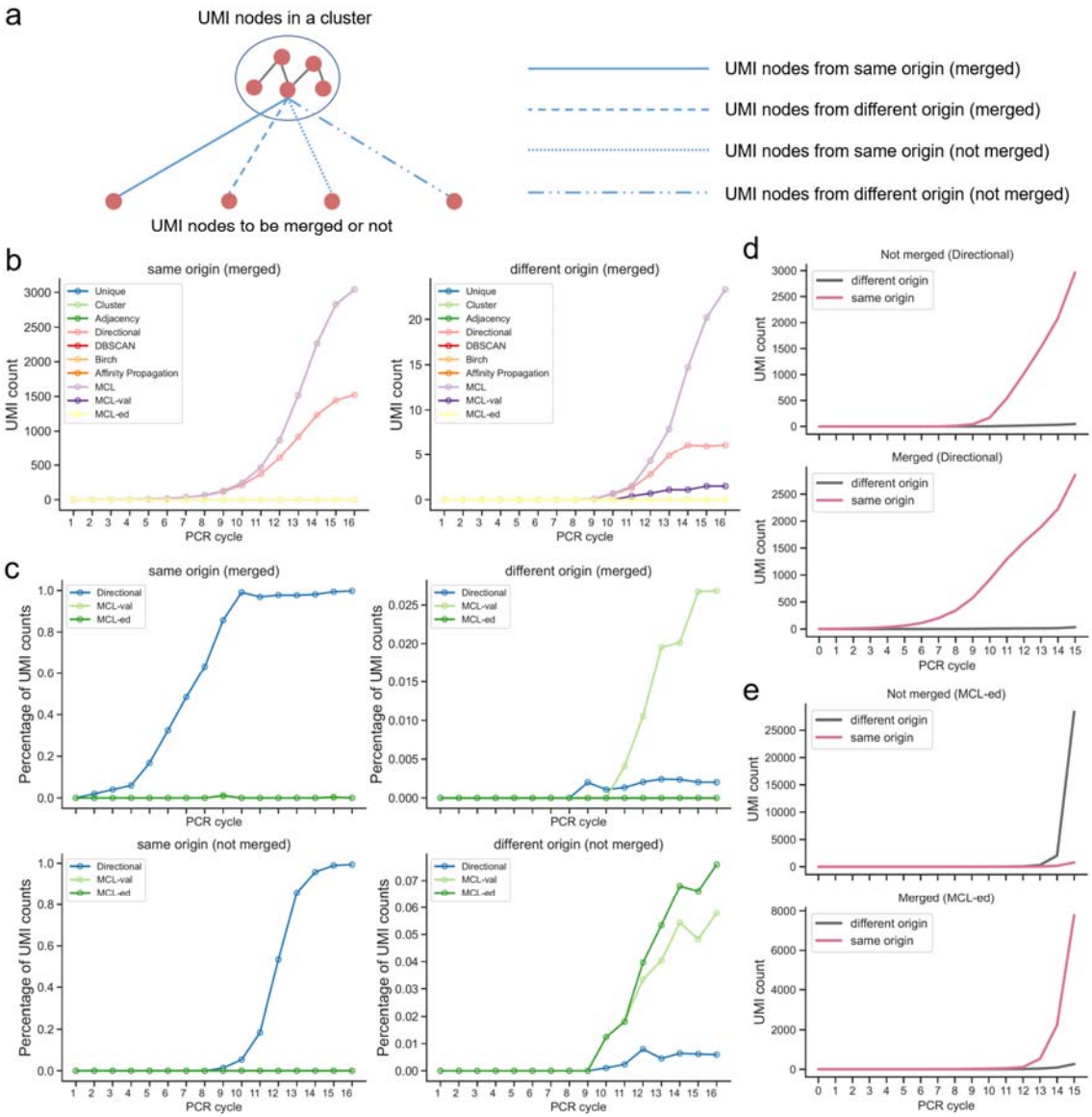


Figure 3. Researching trajectories of UMIs across PCR amplification cycles. **a**, Schematic of merging UMIs into a cluster. Annotating UMIs with their origin is conducive to study their connectivity in a network in which a merge indicates an estimated association between two

UMIs. **b**, Counts of UMIs derived from the same origin to be merged across multiple PCR cycles with different UMI collapsing methods. **c**, Comparison of counts of merged or not merged UMIs derived from the same or different origin between the MCL-val/MCL-ed and Directional methods. **d** and **e**, Counts of merged or not merged UMIs derived from the same or different origin using the Directional (**d**) and MCL-ed (**e**) methods.

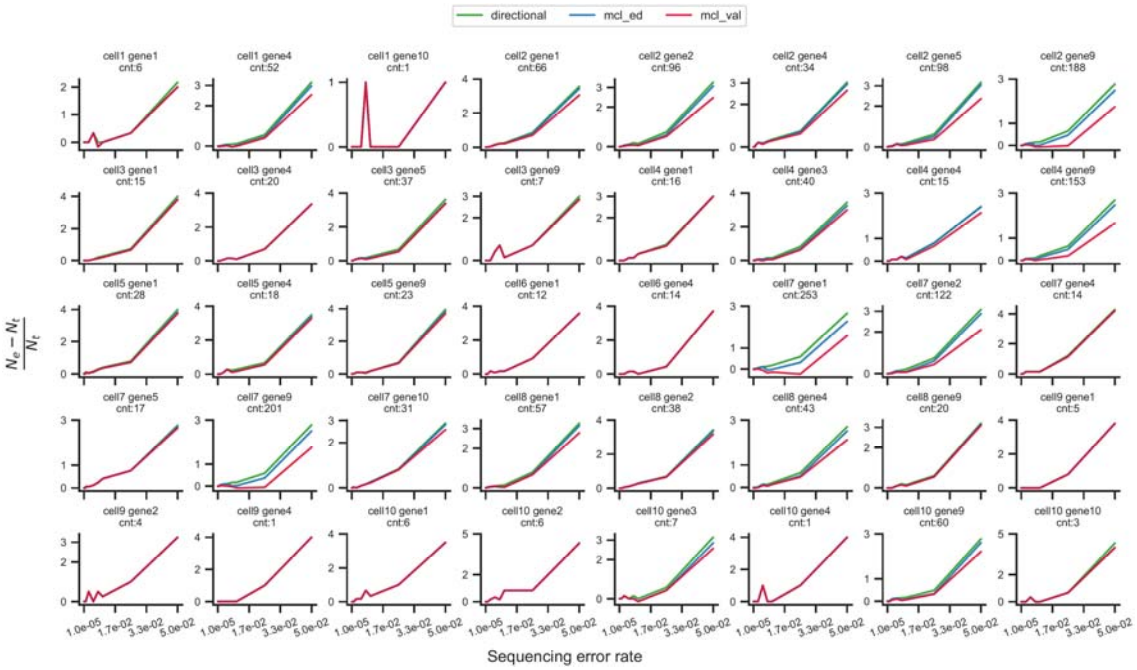


Figure 4. Comparison of UMI counting performance between mclUMI (MCL-ed and MCL-val) and the UMI-tools *Directional* method on single-cell sequencing data simulated by Resimpy at multiple sequencing error rates. Each subplot shows the UMI collapsing result for a certain cell and a gene. cnt: the count of a gene expressed at a cell.

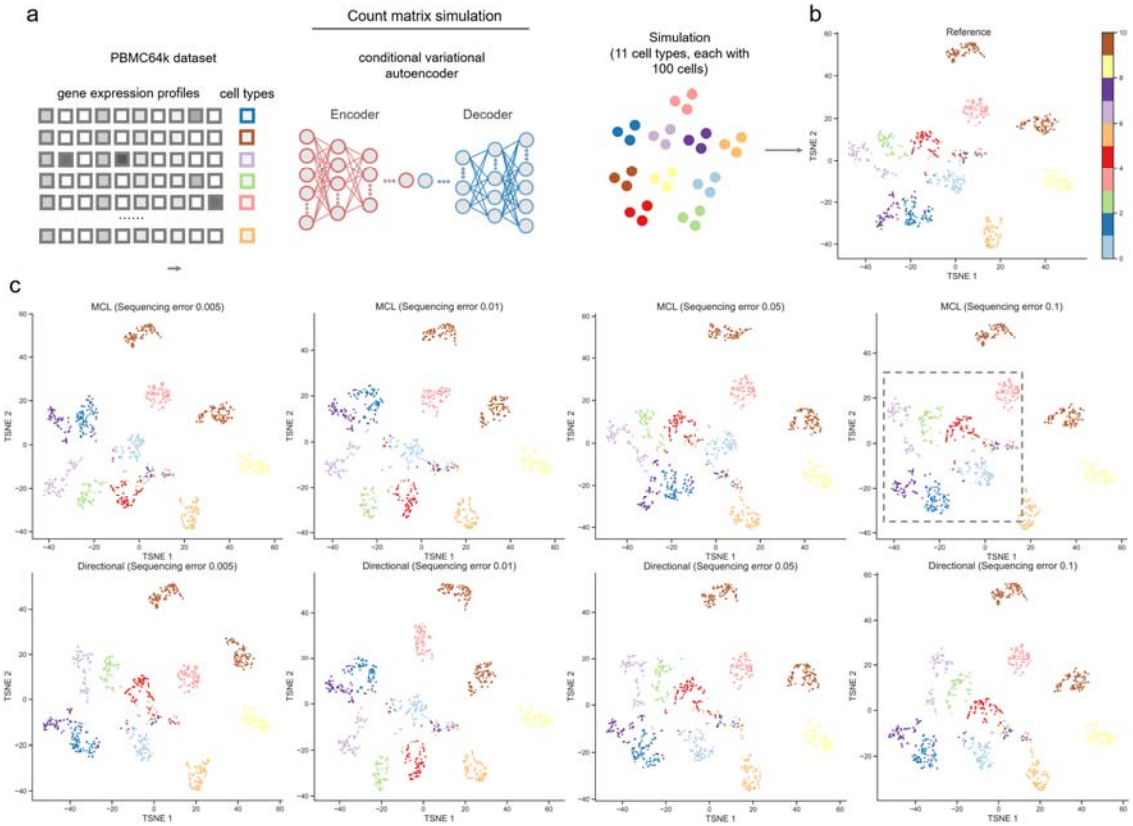


Figure 5. Clustering of deduplicated UMIs with single-cell sequencing data. A count matrix containing 100 cells for each of the 11 cell types is generated using our in-house count matrix simulator based on autoencoders. The resulting count matrix is then visualised by the TSNE method as reference. To estimate the counts per cell-by-gene type by UMI collapsing methods, we simulated the reads according to the count matrix through PCR amplification and sequencing. Subsequently, the sequenced reads with ground truth are respectively deduplicated by the Directional and MCL methods at 0.005, 0.01, 0.05, and 0.1 sequencing error rates. Ultimately, deduplicated counts for each cell-by-gene type are projected by TSNE.

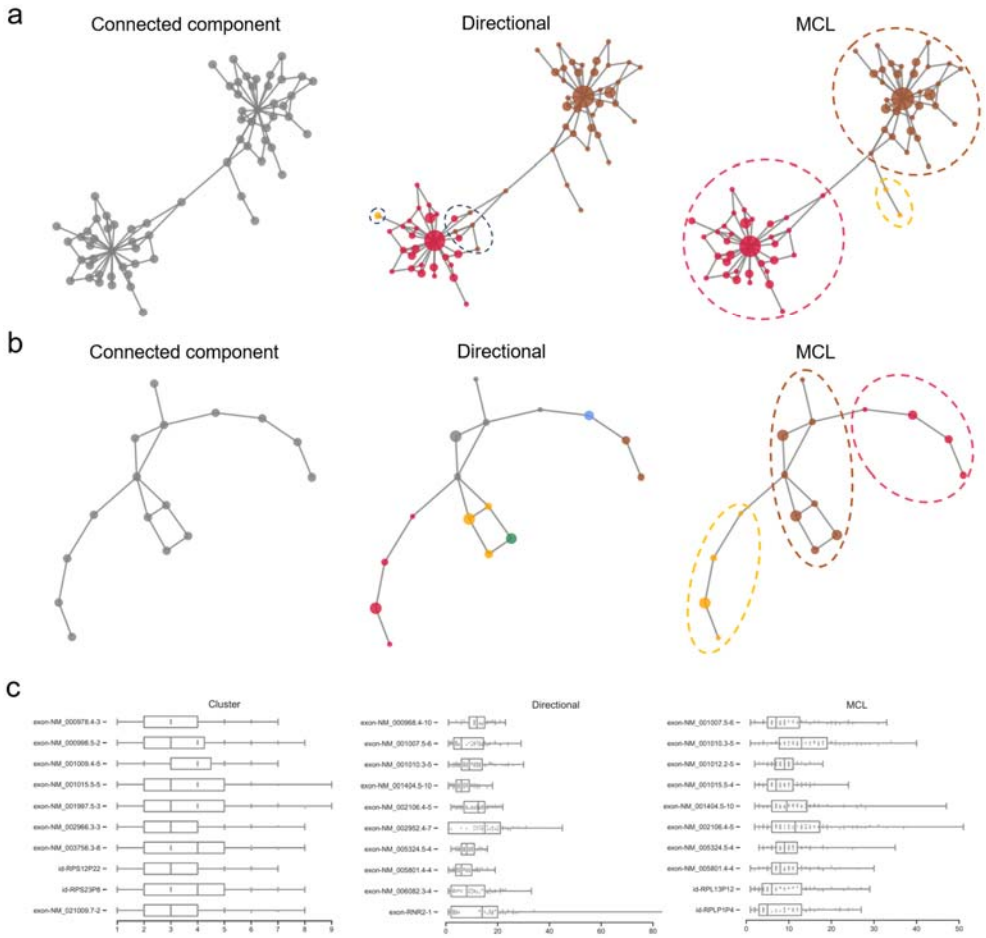


Figure 6. Quality evaluation and biological application of deduplicated UMI counts. **a** and **b**, UMI clusters observed at two example genomic loci using simulated data (**a**) and iCLIP data (**b**) [32], respectively, are discovered from connected components by the Directional and MCL methods. It contained 1,175,027 reads with 20,683 raw unique UMI sequences and 12,047 genomic positions tagged by running the get_bundles method of UMI-tools. Each node in the graphs represents a unique UMI. The node size is enlarged as the count of the unique UMIs increases. Different clusters are shown in different colours. **c**, Comparison of top 10 highly expressed genes identified by the Cluster, Directional, and MCL methods using a 10X genomics scRNA-seq dataset [33]. It contained 3,553,230 raw reads and left 588,963 reads following STAR (version: v2.7.9a) [46] mapping against GRCh38 and gene annotations using featureCounts (version: v2.0.1) [47]. The 100 barcodes were generated using the whitelist function of UMI-tools.

References

1. Logsdon GA, Vollger MR, Eichler EE. Long-read human genome sequencing and its applications. Nat Rev Genet [Internet]. 2020;21:597–614. Available from: https://doi.org/10.1038/s41576-020-0236-x

2. Stark R, Grzelak M, Hadfield J. RNA sequencing: the teenage years. Nat Rev Genet [Internet]. 2019;20:631–56. Available from: https://doi.org/10.1038/s41576-019-0150-2

3. Hiatt JB, Patwardhan RP, Turner EH, Lee C, Shendure J. Parallel, tag-directed assembly of locally derived short sequence reads. Nat Methods [Internet]. 2010;7:119–22. Available from: https://doi.org/10.1038/nmeth.1416

4. De Luca G, Dono M. The Opportunities and Challenges of Molecular Tagging Next-Generation Sequencing in Liquid Biopsy. Mol Diagn Ther [Internet]. 2021;25:537–47. Available from: https://doi.org/10.1007/s40291-021-00542-6

5. Marx V. How to deduplicate PCR. Nat Methods [Internet]. 2017;14:473–6. Available from: https://doi.org/10.1038/nmeth.4268

6. Andersson D, Kebede FT, Escobar M, Österlund T, Ståhlberg A. Principles of digital sequencing using unique molecular identifiers. Mol Aspects Med [Internet]. 2024;96:101253. Available from: https://www.sciencedirect.com/science/article/pii/S0098299724000128

7. Zhu T, Liao K, Zhou R, Xia C, Xie W. ATAC-seq with unique molecular identifiers improves quantification and footprinting. Commun Biol [Internet]. 2020;3:675. Available from: https://doi.org/10.1038/s42003-020-01403-4

8. Islam S, Zeisel A, Joost S, La Manno G, Zajac P, Kasper M, et al. Quantitative single-cell RNA-seq with unique molecular identifiers. Nat Methods [Internet]. 2014;11:163–6. Available from: https://doi.org/10.1038/nmeth.2772

9. Kinde I, Wu J, Papadopoulos N, Kinzler KW, Vogelstein B. Detection and quantification of rare mutations with massively parallel sequencing. Proceedings of the National Academy of Sciences [Internet]. 2011;108:9530–5. Available from: https://www.pnas.org/doi/abs/10.1073/pnas.1105422108

10. Shiroguchi K, Jia TZ, Sims PA, Xie XS. Digital RNA sequencing minimizes sequence-dependent bias and amplification noise with optimized single-molecule barcodes. Proceedings of the National Academy of Sciences [Internet]. 2012;109:1347–52. Available from: https://www.pnas.org/doi/abs/10.1073/pnas.1118018109

11. Kivioja T, Vähärautio A, Karlsson K, Bonke M, Enge M, Linnarsson S, et al. Counting absolute numbers of molecules using unique molecular identifiers. Nat Methods [Internet]. 2012;9:72–4. Available from: https://doi.org/10.1038/nmeth.1778

12. Lareau CA, Duarte FM, Chew JG, Kartha VK, Burkett ZD, Kohlway AS, et al. Droplet-based combinatorial indexing for massive-scale single-cell chromatin accessibility. Nat Biotechnol [Internet]. 2019;37:916–24. Available from: https://doi.org/10.1038/s41587-019-0147-6

13. Zurek PJ, Knyphausen P, Neufeld K, Pushpanath A, Hollfelder F. UMI-linked consensus sequencing enables phylogenetic analysis of directed evolution. Nat Commun [Internet]. 2020;11:6023. Available from: https://doi.org/10.1038/s41467-020-19687-9

14. Fu GK, Hu J, Wang P-H, Fodor SPA. Counting individual DNA molecules by the stochastic attachment of diverse labels. Proceedings of the National Academy of Sciences [Internet]. 2011;108:9026–31. Available from: https://www.pnas.org/doi/abs/10.1073/pnas.1017621108

15. Sun J, Philpott M, Loi D, Li S, Monteagudo-Mesas P, Hoffman G, et al. Correcting PCR amplification errors in unique molecular identifiers to generate accurate numbers of sequencing molecules. Nat Methods [Internet]. 2024;21:401–5. Available from: https://doi.org/10.1038/s41592-024-02168-y

16. Wu Y, Zhang K. Tools for the analysis of high-dimensional single-cell RNA sequencing data. Nat Rev Nephrol [Internet]. 2020;16:408–21. Available from: https://doi.org/10.1038/s41581-020-0262-0

17. Xu C, Gu X, Padmanabhan R, Wu Z, Peng Q, DiCarlo J, et al. smCounter2: an accurate low-frequency variant caller for targeted sequencing data with unique molecular identifiers. Bioinformatics [Internet]. 2018;35:1299–309. Available from: https://doi.org/10.1093/bioinformatics/bty790

18. Melsted P, Ntranos V, Pachter L. The barcode, UMI, set format and BUStools. Bioinformatics [Internet]. 2019;35:4472–3. Available from: https://doi.org/10.1093/bioinformatics/btz279

19. Clement K, Farouni R, Bauer DE, Pinello L. AmpUMI: design and analysis of unique molecular identifiers for deep amplicon sequencing. Bioinformatics [Internet]. 2018;34:i202–10. Available from: https://doi.org/10.1093/bioinformatics/bty264

20. Parekh S, Ziegenhain C, Vieth B, Enard W, Hellmann I. zUMIs - A fast and flexible pipeline to process RNA sequencing data with UMIs. Gigascience [Internet]. 2018;7:giy059. Available from: https://doi.org/10.1093/gigascience/giy059

21. Pflug FG, von Haeseler A. TRUmiCount: correctly counting absolute numbers of molecules using unique molecular identifiers. Bioinformatics [Internet]. 2018;34:3137–44. Available from: https://doi.org/10.1093/bioinformatics/bty283

22. Peng X, Dorman KS. Accurate estimation of molecular counts from amplicon sequence data with unique molecular identifiers. Bioinformatics [Internet]. 2023;39:btad002. Available from: https://doi.org/10.1093/bioinformatics/btad002

23. Tsagiopoulou M, Maniou MC, Pechlivanis N, Togkousidis A, Kotrová M, Hutzenlaub T, et al. UMIc: A Preprocessing Method for UMI Deduplication and Reads Correction. Front Genet [Internet]. 2021;12. Available from: https://www.frontiersin.org/journals/genetics/articles/10.3389/fgene.2021.660366

24. Smith T, Heger A, Sudbery I. UMI-tools: modeling sequencing errors in Unique Molecular Identifiers to improve quantification accuracy. Genome Res [Internet]. 2017;27:491–9. Available from: http://genome.cshlp.org/content/27/3/491.abstract

25. Liu D. Algorithms for efficiently collapsing reads with Unique Molecular Identifiers. PeerJ [Internet]. 2019;7:e8275. Available from: https://doi.org/10.7717/peerj.8275

26. Enright AJ, Van Dongen S, Ouzounis CA. An efficient algorithm for large-scale detection of protein families. Nucleic Acids Res [Internet]. 2002;30:1575–84. Available from: https://doi.org/10.1093/nar/30.7.1575

27. Van Dongen S. Graph Clustering Via a Discrete Uncoupling Process. SIAM Journal on Matrix Analysis and Applications [Internet]. 2008;30:121–41. Available from: https://doi.org/10.1137/040608635

28. Van Dongen S. Graph clustering by flow simulation. PhD thesis, University of Utrecht. 2000;

29. Assefa AT, Vandesompele J, Thas O. SPsimSeq: semi-parametric simulation of bulk and single-cell RNA-sequencing data. Bioinformatics [Internet]. 2020;36:3276–8. Available from: https://doi.org/10.1093/bioinformatics/btaa105

30. Zheng GXY, Terry JM, Belgrader P, Ryvkin P, Bent ZW, Wilson R, et al. Massively parallel digital transcriptional profiling of single cells. Nat Commun [Internet]. 2017;8:14049. Available from: https://doi.org/10.1038/ncomms14049

31. van der Maaten L, Hinton G. Visualizing Data using t-SNE. Journal of Machine Learning Research [Internet]. 2008;9:2579–605. Available from: http://jmlr.org/papers/v9/vandermaaten08a.html

32. Müller-McNicoll M, Botti V, de Jesus Domingues AM, Brandl H, Schwich OD, Steiner MC, et al. SR proteins are NXF1 adaptors that link alternative RNA processing to mRNA export. Genes Dev [Internet]. 2016;30:553–66. Available from: http://genesdev.cshlp.org/content/30/5/553.abstract

33. Klein AM, Mazutis L, Akartuna I, Tallapragada N, Veres A, Li V, et al. Droplet Barcoding for Single-Cell Transcriptomics Applied to Embryonic Stem Cells. Cell [Internet]. 2015;161:1187–201. Available from: https://www.sciencedirect.com/science/article/pii/S0092867415005000

34. Azad A, Pavlopoulos GA, Ouzounis CA, Kyrpides NC, Buluç A. HipMCL: a high-performance parallel implementation of the Markov clustering algorithm for large-scale networks. Nucleic Acids Res [Internet]. 2018;46:e33–e33. Available from: https://doi.org/10.1093/nar/gkx1313

35. Satuluri V, Parthasarathy S, Ucar D. Markov clustering of protein interaction networks with improved balance and scalability. Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology [Internet]. New York, NY, USA: Association for Computing Machinery; 2010. p. 247–56. Available from: https://doi.org/10.1145/1854776.1854812

36. Ester M, Kriegel H-P, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. AAAI Press; 1996. p. 226–31.

37. Zhang T, Ramakrishnan R, Livny M. BIRCH: an efficient data clustering method for very large databases. Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data [Internet]. New York, NY, USA: Association for Computing Machinery; 1996. p. 103–14. Available from: https://doi.org/10.1145/233269.233324

38. Frey BJ, Dueck D. Mixture Modeling by Affinity Propagation. In: Weiss Y, Schölkopf B, Platt J, editors. Adv Neural Inf Process Syst [Internet]. MIT Press; 2005. Available from: https://proceedings.neurips.cc/paper_files/paper/2005/file/327708dd10d68b1361ad3addbaca01f2-Paper.pdf

39. Schubert E, Sander J, Ester M, Kriegel HP, Xu X. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. ACM Trans Database Syst [Internet]. 2017;42. Available from: https://doi.org/10.1145/3068335

40. Zhang T, Ramakrishnan R, Livny M. BIRCH: A New Data Clustering Algorithm and Its Applications. Data Min Knowl Discov [Internet]. 1997;1:141–82. Available from: https://doi.org/10.1023/A:1009783824328

41. Shang F, Jiao LC, Shi J, Wang F, Gong M. Fast affinity propagation clustering: A multilevel approach. Pattern Recognit [Internet]. 2012;45:474–86. Available from: https://www.sciencedirect.com/science/article/pii/S0031320311002007

42. Sun F. The Polymerase Chain Reaction and Branching Processes. Journal of Computational Biology [Internet]. 1995;2:63–86. Available from: https://doi.org/10.1089/cmb.1995.2.63

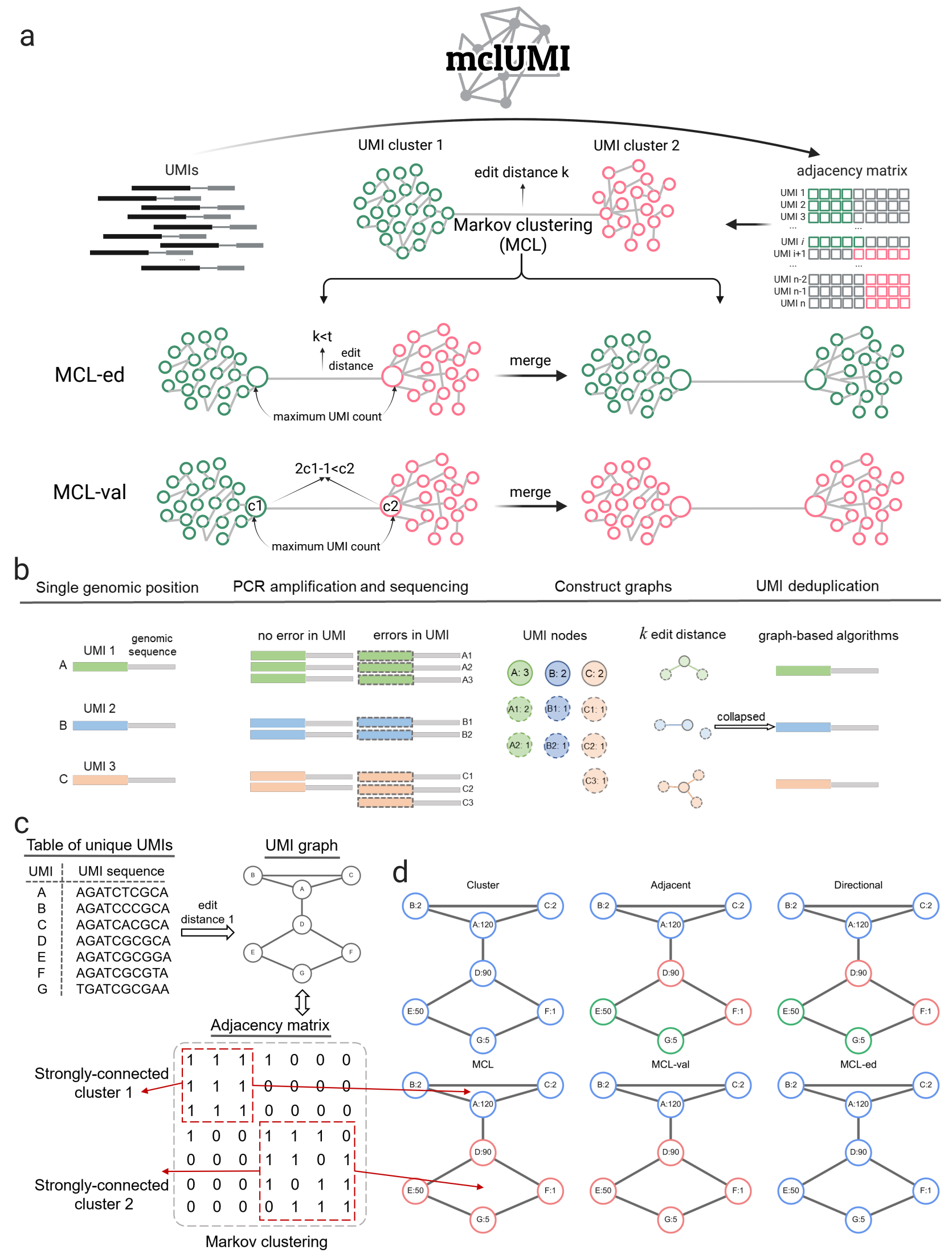
43. Lalam N. Estimation of the reaction efficiency in polymerase chain reaction. J Theor Biol [Internet]. 2006;242:947–53. Available from: https://www.sciencedirect.com/science/article/pii/S0022519306002281

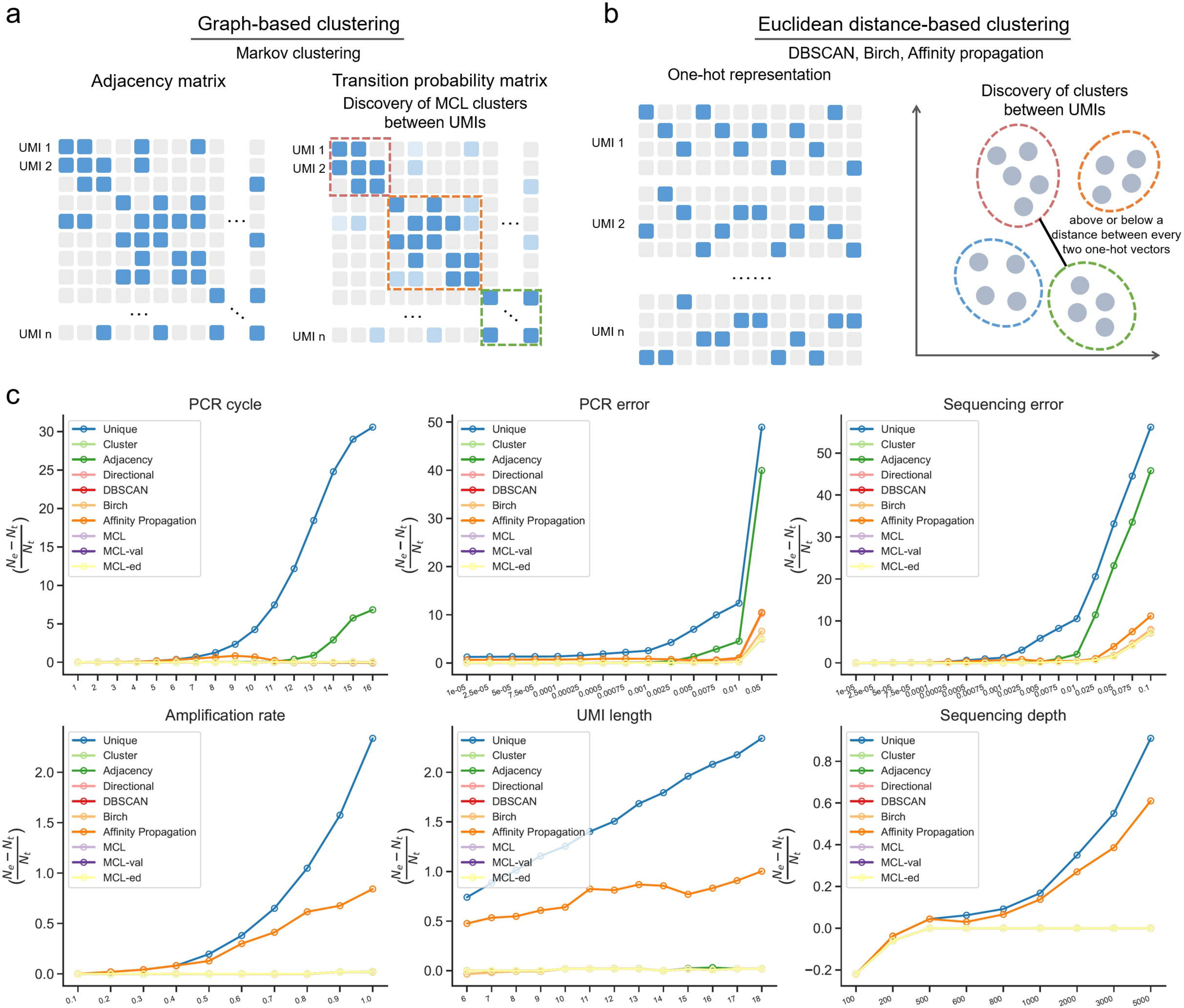
44. Rabadan R, Bhanot G, Marsilio S, Chiorazzi N, Pasqualucci L, Khiabanian H. On Statistical Modeling of Sequencing Noise in High Depth Data to Assess Tumor Evolution. J Stat Phys [Internet]. 2018;172:143–55. Available from: https://doi.org/10.1007/s10955-017-1945-1

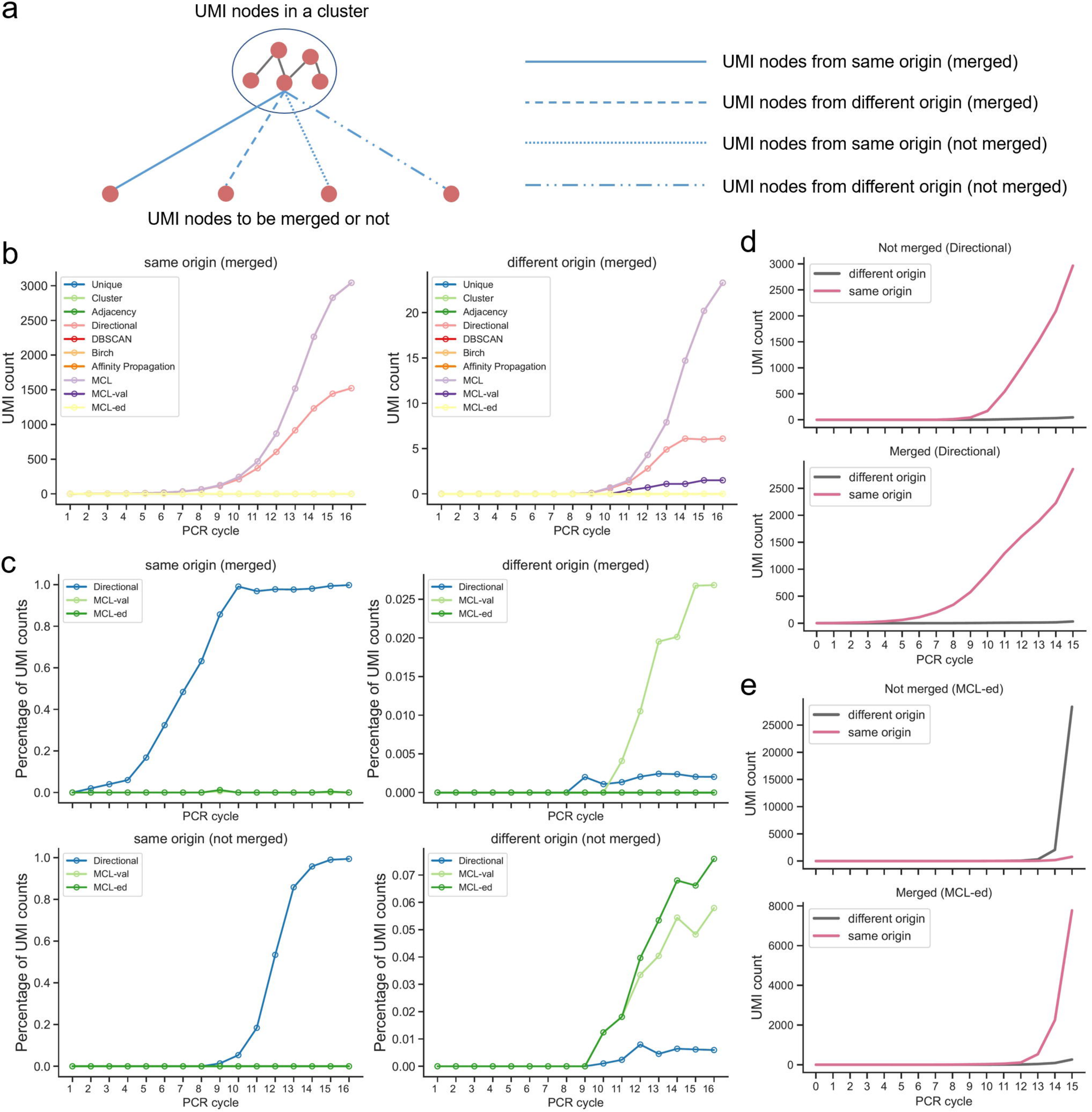
45. Amezquita RA, Lun ATL, Becht E, Carey VJ, Carpp LN, Geistlinger L, et al. Orchestrating single-cell analysis with Bioconductor. Nat Methods [Internet]. 2020;17:137–45. Available from: https://doi.org/10.1038/s41592-019-0654-x

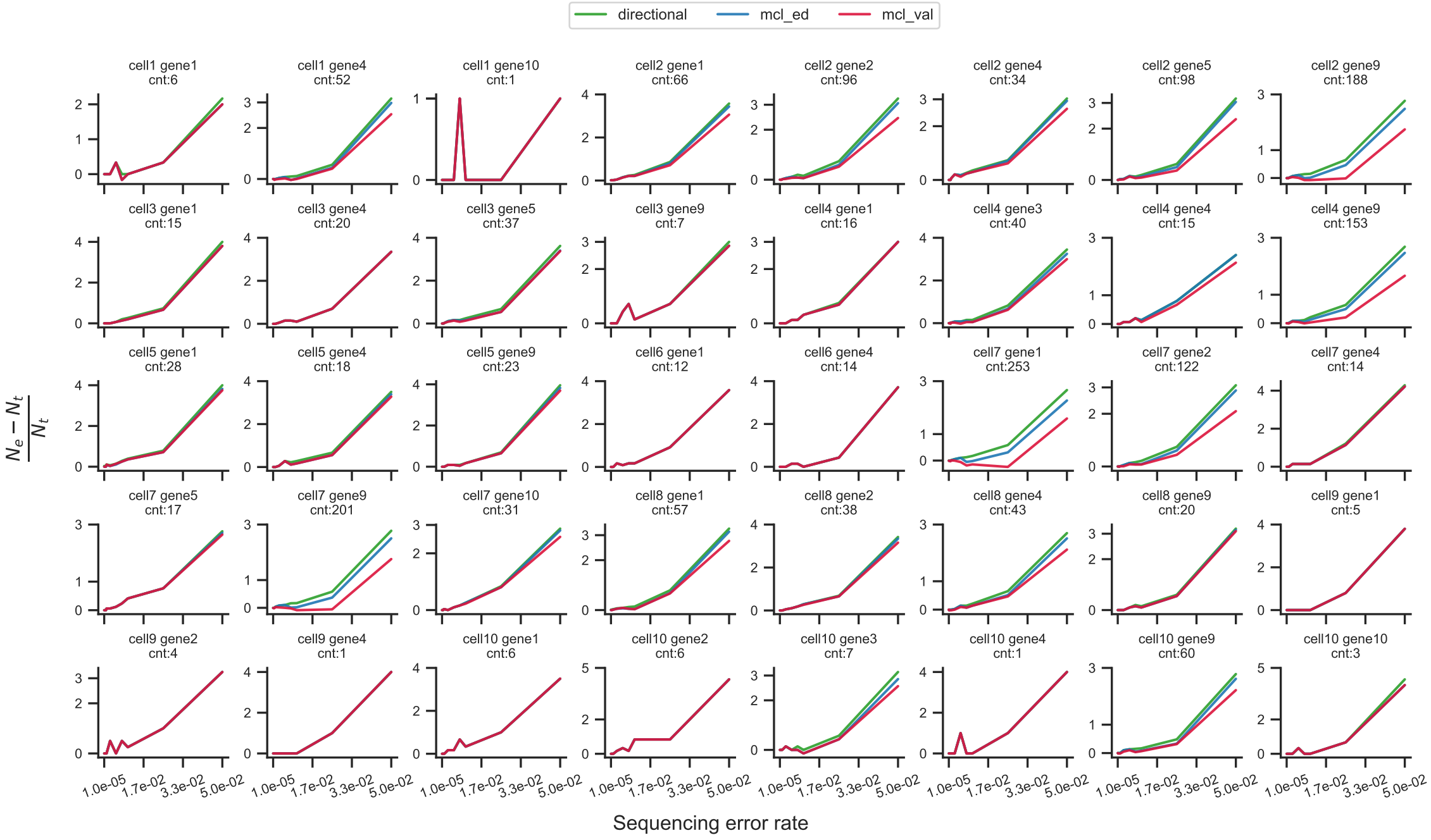
46. Dobin A, Davis CA, Schlesinger F, Drenkow J, Zaleski C, Jha S, et al. STAR: ultrafast universal RNA-seq aligner. Bioinformatics [Internet]. 2012;29:15–21. Available from: https://doi.org/10.1093/bioinformatics/bts635

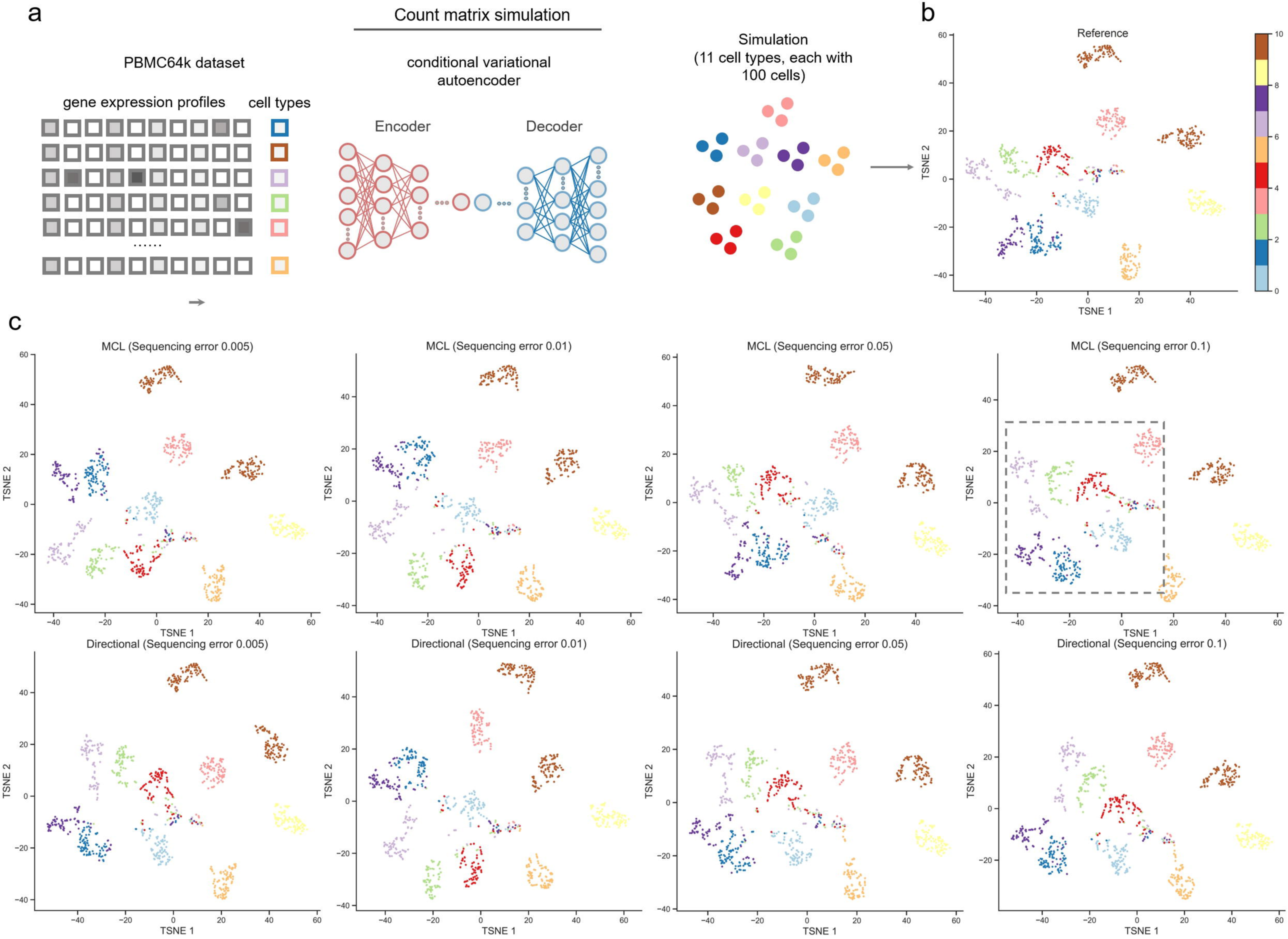
47. Liao Y, Smyth GK, Shi W. featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. Bioinformatics [Internet]. 2013;30:923–30. Available from: https://doi.org/10.1093/bioinformatics/btt656

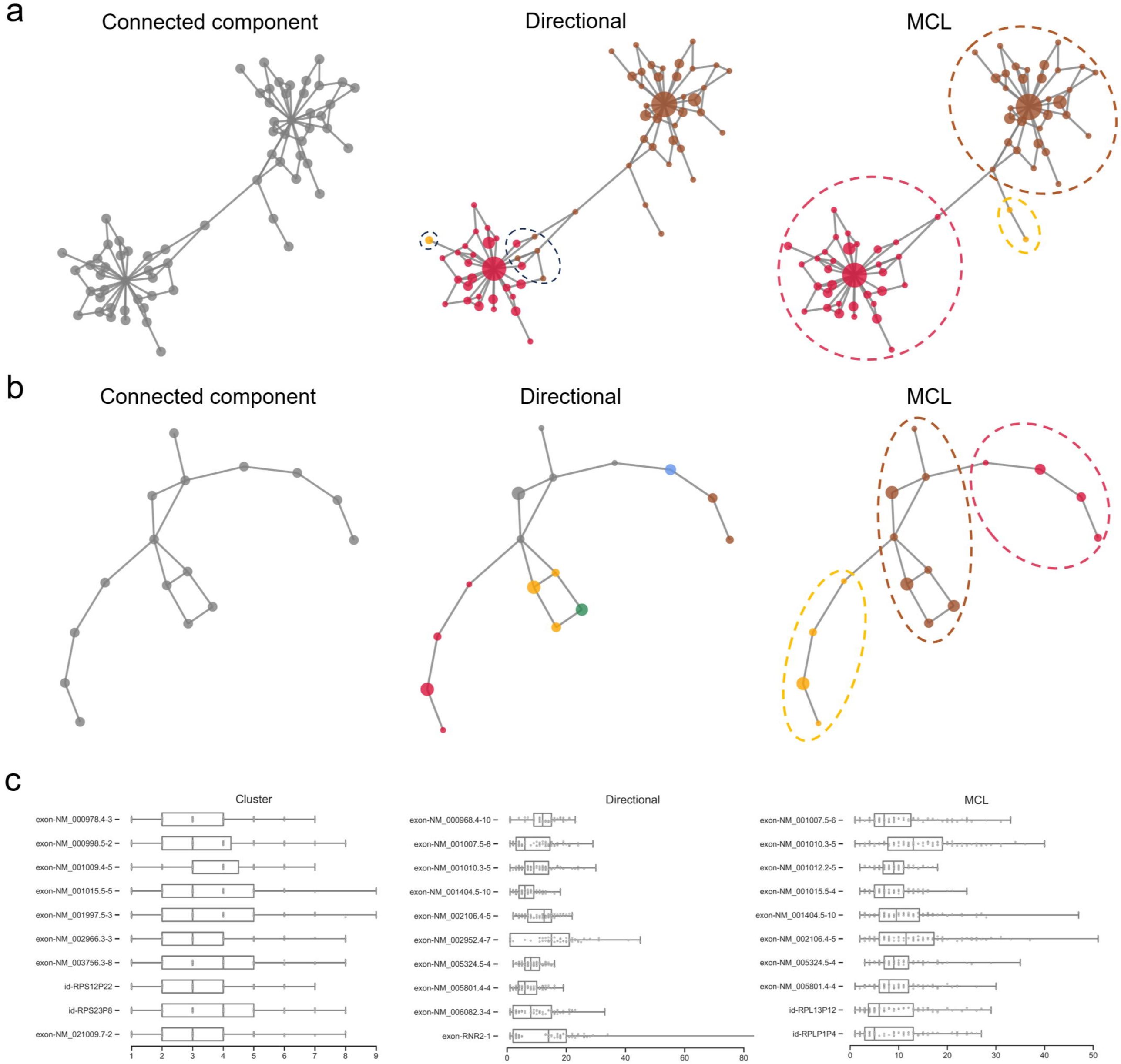

a
mclUMI
UMIs
UMI cluster 1
UMI cluster 2
edit distance k
Markov clustering (MCL)
adjacency matrix
UMI 1
UMI 2
UMI 3
UMI i
UMI i+1
UMI n-2
UMI n-1
UMI n
MCL-ed
k<t
edit distance
maximum UMI count
merge
MCL-val
2c1-1<c2
c1
c2
maximum UMI count
merge
b
Single genomic position
PCR amplification and sequencing
Construct graphs
UMI deduplication
UMI 1
genomic sequence
A
UMI 2
B
UMI 3
C
no error in UMI
errors in UMI
A1
A2
A3
B1
B2
C1
C2
C3
UMI nodes
A: 3
B: 2
C: 2
A1: 2
B1: 1
C1: 1
A2: 1
B2: 1
C2: 1
C3: 1
k edit distance
collapsed
graph-based algorithms
c
Table of unique UMIs
UMI
UMI sequence
A AGATCTCGCA
B AGATCCCGCA
C AGATCACGCA
D AGATCGCGCA
E AGATCGCGGA
F AGATCGCGTA
G TGATCGCGAA
edit distance 1
UMI graph
Adjacency matrix
Strongly-connected cluster 1
Strongly-connected cluster 2
Markov clustering
d
Cluster
Adjacent
Directional
MCL
MCL-val
MCL-ed
B:2
C:2
A:120
D:90
E:50
F:1
G:5


a
Graph-based clustering
Markov clustering
Adjacency matrix
Transition probability matrix
Discovery of MCL clusters between UMIs
UMI 1
UMI 2
UMI n
b
Euclidean distance-based clustering
DBSCAN, Birch, Affinity propagation
One-hot representation
Discovery of clusters between UMIs
above or below a distance between every two one-hot vectors
c
PCR cycle
PCR error
Sequencing error
Amplification rate
UMI length
Sequencing depth
Unique
Cluster
Adjacency
Directional
DBSCAN
Birch
Affinity Propagation
MCL
MCL-val
MCL-ed
$\left(\frac{N_e - N_t}{N_t}\right)$


a
UMI nodes in a cluster
UMI nodes to be merged or not
UMI nodes from same origin (merged)
UMI nodes from different origin (merged)
UMI nodes from same origin (not merged)
UMI nodes from different origin (not merged)
b
same origin (merged)
different origin (merged)
UMI count
PCR cycle
Unique
Cluster
Adjacency
Directional
DBSCAN
Birch
Affinity Propagation
MCL
MCL-val
MCL-ed
c
same origin (merged)
different origin (merged)
same origin (not merged)
different origin (not merged)
Percentage of UMI counts
PCR cycle
Directional
MCL-val
MCL-ed
d
Not merged (Directional)
Merged (Directional)
UMI count
PCR cycle
different origin
same origin
e
Not merged (MCL-ed)
Merged (MCL-ed)
UMI count
PCR cycle
different origin
same origin


directional
mcl_ed
mcl_val
cell1 gene1 cnt:6
cell1 gene4 cnt:52
cell1 gene10 cnt:1
cell2 gene1 cnt:66
cell2 gene2 cnt:96
cell2 gene4 cnt:34
cell2 gene5 cnt:98
cell2 gene9 cnt:188
cell3 gene1 cnt:15
cell3 gene4 cnt:20
cell3 gene5 cnt:37
cell3 gene9 cnt:7
cell4 gene1 cnt:16
cell4 gene3 cnt:40
cell4 gene4 cnt:15
cell4 gene9 cnt:153
cell5 gene1 cnt:28
cell5 gene4 cnt:18
cell5 gene9 cnt:23
cell6 gene1 cnt:12
cell6 gene4 cnt:14
cell7 gene1 cnt:253
cell7 gene2 cnt:122
cell7 gene4 cnt:14
cell7 gene5 cnt:17
cell7 gene9 cnt:201
cell7 gene10 cnt:31
cell8 gene1 cnt:57
cell8 gene2 cnt:38
cell8 gene4 cnt:43
cell8 gene9 cnt:20
cell9 gene1 cnt:5
cell9 gene2 cnt:4
cell9 gene4 cnt:1
cell10 gene1 cnt:6
cell10 gene2 cnt:6
cell10 gene3 cnt:7
cell10 gene4 cnt:1
cell10 gene9 cnt:60
cell10 gene10 cnt:3
$\frac{N_e - N_t}{N_t}$
1.0e-05
1.7e-02
3.3e-02
5.0e-02
Sequencing error rate


a
Count matrix simulation
PBMC64k dataset
gene expression profiles
cell types
.....
conditional variational autoencoder
Encoder
Decoder
Simulation
(11 cell types, each with 100 cells)
b
Reference
TSNE 1
TSNE 2
c
MCL (Sequencing error 0.005)
MCL (Sequencing error 0.01)
MCL (Sequencing error 0.05)
MCL (Sequencing error 0.1)
Directional (Sequencing error 0.005)
Directional (Sequencing error 0.01)
Directional (Sequencing error 0.05)
Directional (Sequencing error 0.1)


a
Connected component
Directional
MCL
b
Connected component
Directional
MCL
C
Cluster
exon-NM_000978.4-3
exon-NM_000998.5-2
exon-NM_001009.4-5
exon-NM_001015.5-5
exon-NM_001997.5-3
exon-NM_002966.3-3
exon-NM_003756.3-8
id-RPS12P22
id-RPS23P8
exon-NM_021009.7-2
1 2 3 4 5 6 7 8 9
Directional
exon-NM_000968.4-10
exon-NM_001007.5-6
exon-NM_001010.3-5
exon-NM_001404.5-10
exon-NM_002106.4-5
exon-NM_002952.4-7
exon-NM_005324.5-4
exon-NM_005801.4-4
exon-NM_006082.3-4
exon-RNR2-1
0 20 40 60 80
MCL
exon-NM_001007.5-6
exon-NM_001010.3-5
exon-NM_001012.2-5
exon-NM_001015.5-4
exon-NM_001404.5-10
exon-NM_002106.4-5
exon-NM_005324.5-4
exon-NM_005801.4-4
id-RPL13P12
id-RPLP1P4
0 10 20 30 40 50