Error-Oblivious Sample Preparation with Digital Microfluidic Lab-on-Chip

Sudip Poddar, Robert Wille, *Senior Member, IEEE*,
Hafizur Rahaman, *Senior Member, IEEE*, and Bhargab B. Bhattacharya, *Fellow, IEEE*

Abstract—**Microfluidic chips are now being increasingly used for fast and cost-effective implementation of biochemical protocols. Sample preparation involves dilution and mixing of fluids in certain ratios, which are needed for most of the protocols. On a digital microfluidic biochip (*DMFB*), these tasks are usually automated as a sequence of droplet mix-split steps. In the most widely used (1:1) mix-split operation for *DMFB*s, two equal-volume droplets are mixed followed by a split operation, which, ideally, should produce two daughter-droplets of equal volume (balanced splitting). However, because of uncertain variabilities in fluidic operations, the outcome of droplet-split operations often becomes erroneous, i.e., they may cause unbalanced splitting. As a result, the concentration factor *(CF)* of each constituent fluid in the mixture may become erroneous during sample preparation. All traditional approaches aimed to recover from such errors deploy on-chip sensors to detect possible volumetric imbalance, and adopt either checkpointing-based rollback or roll-forward techniques. Most of them suffer from significant overhead in terms of assay-completion time, reactant-cost, and uncertainties in termination due to randomly occurring split-errors. In this paper, we propose a new approach to accurate dilution preparation on a *DMFB* that is *oblivious* to volumetric split-errors. It does not need any sensor and can handle multiple split-errors, deterministically. The proposed method is customized for each target-*CF* based on the criticality of split-errors in each mix-split step. Simulation experiments on various test-cases demonstrate the effectiveness of the proposed method.**

I. Introduction

Digital Microfluidic Biochips (*DMFBs*), as one possible plattform for Labs-on-chip (*LoC*), received significant attention of the *VLSI CAD* community over the last few years due to their versatile applications in biochemical domains such as point-of-care (PoC) testing, drug discovery, or high throughput *DNA* sequencing, to name a few [1], [2]. A *DMFB* is a coin-sized device capable of performing several biochemical protocols by manipulating discrete fluid-droplets on the top of an electrostatically-controlled *2D*-array of tiny electrodes. These chips are likely to replace expensive and bulky equipment currently used in hospitals, pathological and research laboratories as they offer more convenience in terms of high throughput, portability, automation, low reagent-consumption, fast reaction-time as well as low energy-consumption.

Based on the principle of electrowetting-on-dielectric (*EWOD*) [3], discrete-sized droplets (with volume in the nano or pico-liter range) can be induced to perform various on-chip fluidic operations such as dispensing, transportation, mixing, splitting, incubation, or sensing by applying a time-varying voltage signal to the electrodes (actuation sequence). By appropriate synthesis of a given protocol, the actuation sequence needed for its execution can efficiently be generated [4]–[7].

One of the most important components of a bio-protocol is sample preparation, which includes, among others, the task of preparing fluid-dilutions and reagent-mixtures in certain ratios. The *DMFB*-technology offers a very convenient platform for automating sample preparation [8]–[12], where a desired target concentration factor (*CF*) of a sample or a mixing-ratio of reagents is produced by performing a sequence of droplet-mixing and splitting operations. Note that the *CF* of a fluid-sample indicates the volumetric ratio of the corresponding raw (pure) sample in a mixture of fluids, i.e., $0 \leq CF \leq 1$. Also, the underlying mix-split sequence is usually abstracted using a directed graph called mix-split graph or task-graph [8], [9].

More precisely, in the discrete mixing model, a droplet with integral volume v_1 units with *CF* = c_1 is allowed to mix with another droplet with integral volume v_2 units with *CF* = c_2. When the mixing is completed, the resulting droplet is split into two equal-size daughter-droplets (i.e., balanced splitting), each with volume $\frac{(v_1+v_2)}{2}$, and the resulting *CF* becomes $\frac{(c_1 \times v_1)+(c_2 \times v_2)}{v_1+v_2}$. For the most widely used (1:1) mixing model, v_1 = v_2 = 1, and the corresponding unit volume is denoted as 1X. The objective of sample preparation is to achieve the desired *CF* for each of the constituents in a fluid mixture and to minimize the number of mix-split operations, reactant-cost, or waste production [8]–[14].

In the (1:1) mixing model, every mix-split operation is performed between two unit volume 1X droplets yielding a 2X droplet. Note that the resulting *CF* obtained through a sequence of (1:1) mix-split steps will be accurate only when the assumption of balanced splitting holds, that is, a droplet of volume 2X is always split into two daughter-droplets of

Manuscript received October 18, 2017; revised March 02, 2018 and May 10; accepted June 29, 2018. This paper was recommended by Associate Editor Tsung-Yi Ho.

S. Poddar, and B. B. Bhattacharya are with the Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata, India 700108. E-mail: {sudippoddar2006, bhargab.bhatta}@gmail.com. The work of S. Poddar is supported, in part, by the CSIR Research Associateship, India. The work of B. B. Bhattacharya was supported by INAE Chair Professorship and by the DST SERB Research Grant No. EMR/2016/005977.

H. Rahaman is with the School of VLSI Technology, Indian Institute of Engineering Science and Technology, Shibpur, India 711103. E-mail: hafizur@vlsi.iiests.ac.in.

R. Wille is with the Institute for Integrated Circuits, Johannes Kepler University Linz, Austria. E-mail: robert.wille@jku.at.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier xx.xxxx/TCAD.2018.xxxxxxx

volume 1X each (i.e., two droplets of the same size). However, in *DMFB*, the outcome of an on-chip split operation is highly dependent on the electrode pitch, the placement of the 2X-size droplet on the electrodes before splitting, its volume, the contact angle, which is altered by the *EWOD*-scheme, and the concurrency of actuation-voltage switching. Moreover, several physical and functional faults may occur in *DMFB*s even after declaring the chip as defect-free after manufacturing test [15]; they may arise due to transient shorts between neighbouring electrodes [16], dielectric breakdown, or charge trapping [17]. These faults are likely to affect the outcome of split-operations conducted on the chip [18] and might as well lead to unbalanced split of droplets and, hence, split-errors.

Unbalanced-split errors, described above, obviously pose a significant threat to sample preparation. In fact, many real-life biomedical applications such as clinical diagnostics, DNA analysis, or drug design rely on high precision of *CF*-values of constituents in a fluid mixture, i.e., the outcome of the assay operations cannot be negotiated [19]–[21]. Thus, when the error in *CF* caused by unbalanced splits exceeds a certain threshold value, the obtained sample is rendered useless.

The above problem has so far been addressed by several error-recovery procedures, which are based on either re-execution (rollback) [22]–[27], or roll-forward [18]. All these solutions, however, come with significant drawbacks. First, they all need a cyber-physical *DMFB* with integrated on-chip sensors and a feedback controller in order to continuously monitor the outcome of mix-split modules and to send the corresponding feedback signals to the controller in a timely manner. Besides this hardware overhead, error-detection and subsequent re-execution compromise badly with assay-completion time. In many applications such as flash-chemistry (a powerful tool for drug discovery, clinical diagnosis, and novel material synthesis), the reaction time lies in the interval of milliseconds to seconds [24], [28] and therefore, faster error-recovery techniques are needed. Finally, the aforementioned rollback approaches rely on the fact that "re-execution is always possible" (which might not be the case, e.g., when not enough backup samples are available or when error occurs again during re-execution). This is discussed in more detail later in Section II-C.

In this work, we propose a fundamentally different approach to solve the problem of reliable dilution preparation with *DMFB*s. Instead of constantly sensing for possible occurrence of split-errors and re-executing the affected portion of the task-graph, a *first-try-right* scheme is proposed. To this end, we first investigate the effect of all volumetric split-errors that may arise in different combinations to the target-*CF*. We observe that some combinations of split-errors may cancel out the unwanted effects, and thus be allowed to achieve the desired *CF* within a given error-tolerance. These observations eventually lead to the proposed "sensor-free" methodology, which determines a sequence of mix-split operations that leads to the production of the target-droplet with the desired *CF*. In other words, the proposed method is oblivious to split-errors as there is no need for monitoring intermediate results and re-executing a portion of the assay.

The proposed method aims to achieve several significant advantages for sample preparation on *DMFB*s and yields a solution, which:

- implements a first-try-right scheme, i.e., the desired target-*CF* is produced correctly without the need for any re-execution;
- is oblivious to split-errors, and hence, no on-chip sensors are needed; the actions taken will be the same whether or not any split-error has occurred;
- can handle a large class of multiple split-errors (in contrast to previously proposed approaches, where a limited number of errors were considered);
- investigates the nature of errors in the target-*CF* when the larger or smaller daughter-droplets are selected following different erroneous mix-split steps;
- does not require any online resynthesis effort as proposed in previous cyber-physical approaches, which may increase assay-completion time [23]; and
- does not need to use any on-chip electrodes for retaining backup droplets, which may create additional routing constraints to assay-droplets.

The rest of the paper is organized as follows: Section II reviews the background on sample preparation as well as errors which may occur during this process. Afterwards, the general idea of this work is illustrated by means of an example in Section III. Based on that, Section IV summarizes our observations of the effects on split-errors on target-*CF*, which eventually leads to the proposed methodology described in Section V. Finally, Section VI reports experimental results and the paper is concluded in Section VII.

II. Sample Preparation Using DMFBs

This section provides a more detailed review of the basics on sample preparation using *DMFB*s as well as the most frequently occurring errors during this process.

A. Sample Preparation

Since most of the collected samples cannot be directly utilized for biochemical reactions in a laboratory, some type of preparation is required in almost every assay. For instance, in the dilution problem, the sample fluid has to be prepared with a desired-*CF* with $0 < CF < 1$. More precisely, we start with the raw sample (*CF* = 1) and a buffer solution (*CF* = 0), provided as droplets, and perform a sequence of (1:1) mixing and splitting operations in order to produce the desired target-*CF* [8], [9]. The sequence of mix-split operations to be conducted can be conveniently represented in terms of a directed mix-split graph, where each node denotes the *CF* of the fluid droplet produced by a (1:1) mix-split operation, and the two incoming edges represent the droplets to be mixed. The outgoing edge from a node represents the split-droplets to be used for subsequent mix-operations [8], [9].

Note that the depth of the *mix-split graph/tree*, which generally denotes the accuracy level (n) of a particular *CF*, depends on a user-defined parameter "τ" (error-tolerance limit), where $0 < \tau < 1$. In order to bound the concentration error in the target-*CF* within the error-tolerance

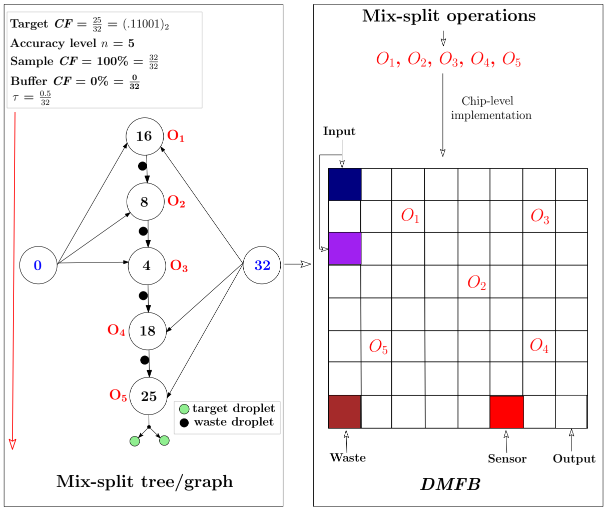


Fig. 1. Mix-split graph for generating target-*CF*=$\frac{25}{32}$ of accuracy level 5. Only the numerator component of each *CF*-value is shown in the diagram; denominator = 2^5 = 32.

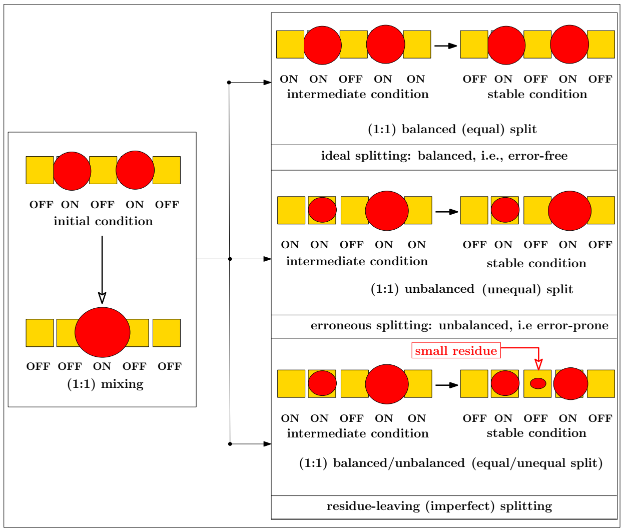


Fig. 2. Ideal and erroneous split operations.

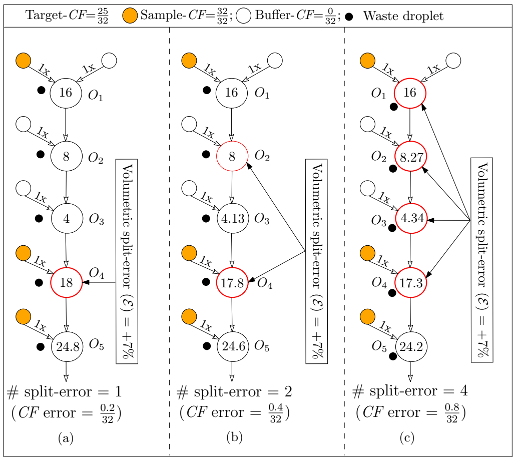

Target-CF=$\frac{25}{32}$
Sample-CF=$\frac{32}{32}$;
Buffer-CF=$\frac{0}{32}$;
Waste droplet
1x
16 O_1
8 O_2
4 O_3
18 O_4
24.8 O_5
Volumetric split-error (ε) = +7%
# split-error = 1
(CF error = $\frac{0.2}{32}$)
(a)
16 O_1
8 O_2
4.13 O_3
17.8 O_4
24.6 O_5
Volumetric split-error (ε) = +7%
# split-error = 2
(CF error = $\frac{0.4}{32}$)
(b)
O_1 16
O_2 8.27
O_3 4.34
O_4 17.3
O_5 24.2
Volumetric split-error (ε) = +7%
# split-error = 4
(CF error = $\frac{0.8}{32}$)
(c)

Fig. 3. Effect of volumetric split-errors to the target-*CF*=$\frac{25}{32}$ for accuracy level 5.

limit (τ) = $\frac{1}{2^{n+1}} = \frac{0.5}{2^n}$ [8], [9], each *CF* is represented as an n-bit binary fractional number $\frac{x}{2^n}$, where $x \in \mathbb{N}$, $0 \leq x \leq 2^n$, and $n \in \mathbb{N}$ [29]. However, for the sake of pictorial clarity, we will represent, each *CF*-value, for a given n, by its numerator only, i.e., we will write $CF = x$ in all figures, instead of $\frac{x}{2^n}$. The value of each *CF* is represented as the label of the corresponding node in the *mix-split graph/tree*.

Example 1. *Fig. 1 shows the mix-split graph for preparing the dilution with CF =* $\frac{25}{32}$ *of a sample following the twoWayMix-algorithm [8]. Note that edges of the mix-split graph determine the order of execution of each mix-split step.*

B. Errors in DMFB

We discuss below some common errors that may occur during sample preparation with *DMFB*s, and their effects.

1) Dispensing Error: Source droplets (i.e., sample or buffer droplets) are injected to a biochip from the input reservoirs placed around its boundary, and volumetric error may occur at the time of dispensing.

One can simply correct the volume of the dispensed droplets by simply returning the erroneous droplet back to the source reservoir and re-dispensing again. Moreover, there also exist other accurate droplet-emission mechanisms [3], [30], [31], which can be used to dispense precise volume of droplets from the reservoirs. Since reliable dispensing solutions are already available, we have not considered this type of error in the remainder of our work.

2) Volumetric Split Error: In general, a split may be balanced, unbalanced, or residue-leaving (imperfect) as illustrated in Fig 2. In a balanced (unbalanced) split, two daughter-droplets are created with equal (unequal) volumes. Sometimes, a small residue is left on the middle electrode during a split operation while producing two equal-/unequal-volume droplets [32], [33]. Such operations not only make the volume of the split-droplets erroneous but also may cause cross-contamination because of the left-over residue [34].

It has been observed experimentally that most of the errors are caused by volumetric imbalance (approx. 80% of all errors), and that these errors may cause a difference of up to 7% in the volume [18], [35]. Note that several mix-split steps are needed during sample preparation and each split operation is a potential source of volumetric imbalance.

Example 2. *Consider again the mix-split graph of Fig. 1 for preparing the target-CF=*$\frac{25}{32}$*. Note that a single volumetric split-error of magnitude 7% in one of the operations in the mix-split graph of Fig. 3 (a) produces a CF-error =* $\frac{0.2}{32}$*. Moreover, if two or even four volumetric split-errors of magnitude 7% occur, the* CF*-error rapidly grows to* $\frac{0.4}{32}$ *and* $\frac{0.8}{32}$ *as illustrated in Fig. 3 (b) and Fig. 3 (c), respectively.*

3) Critical/non-critical set of errors: A set of volumetric split-errors is called *non-critical*, if the produced concentration error in the target-*CF* is $<\frac{0.5}{2^n}$, i.e., bounded by the inherent accuracy level determined by the choice of n, in spite of the occurrence of such multiple errors; otherwise they are termed as *critical* error. The sign of the error $+$ $(-)$ indicates that the larger (smaller) daughter-droplet has been used in the following mixing-path. For example for accuracy level n = 5, the *CF*-error is bounded by $\frac{0.5}{32}$, and hence the errors in the mix-split operations $\{O_1, O_2, O_3, O_4\}$ represent a critical set of errors (since *CF*-error = $\frac{0.8}{32}$), whereas those in $\{O_4\}$ and $\{O_2, O_4\}$ are non-critical errors (Fig. 3).

As observed from Example 2, the impact of the volumetric split-errors on the target-*CF* strictly depends on i) the magnitude and sign of the split-error (ϵ), and ii) its number of occurrences. In this work, we assume that the magnitude of a volumetric split-error is at most 7% [18], [35], and they may occur in any mix-split step of the mixing-path.

C. Currently Proposed Solutions

Several sample-preparation algorithms for *DMFB*s have been reported with different optimization objectives [8]–[11], [38]. However, none of them addresses the impact of split-errors. Some of them attempt to recover the desired *CF* by re-executing a certain portion of an assay using pre-stored backup droplets [22]–[27], when an error is sensed. These approaches assume that the error-recovery operations are error-free [22].

In other error-recovery approaches proposed in [24], [26], all possible sequences of operations are pre-stored in the memory (including those needed for error-recovery for a number of errors) prior to the actual execution of the assay. By this, a limited number of errors (# of errors <= 2) can be handled – at the cost of large memory usage. In order to reduce the cost of the error-recovery operations, a certain portion of the assay may be re-executed [23], [25] for error-recovery using the previously stored backup droplets. However, such on-chip backup droplets require extra storage space and may create additional routing constraints to other active droplets.

A probabilistic approach to volumetric split-error correction has been proposed in [27], where the split operation (involving the erroneous operation) is repeated up to a certain number of times to fix the error. But such scheme requires additional time and the error may occur again after attempting the split operation a certain number of times.

Recently, a roll-forward error-correcting method was proposed for cancelling volumetric split-errors during sample preparation [18]. In spite of discarding the erroneous droplets as waste droplets, they are used in a parallel mirror-path of the *task-graph* in order to correct the effect of volumetric split-errors on the target-*CF* in a deterministic fashion.

Error recovery in Micro-Electrode-Dot-Array (*MEDA*) based *DMFB*s is also recently studied [36]. However, like all traditional rollback approaches, this method also increases the cost of error-recovery in both space and time by reconfiguring the biochip dynamically using the real-time data provided by the sensor. More recently, another hybrid error-recovery approach (rollback and droplet-volume regulation based strategy) was proposed for *MEDA*-based *DMFB* chips [37]. However, it increases assay-completion time as error-detection operation is needed after mix-split steps.

The characteristics and scope of the proposed error-recovery method for *DMFB*s in contrast to prior approaches are summarized in Table I.

III. General Idea of the Proposed Solution

In this work, we propose an alternative scheme for reliable sample preparation on *DMFB*s. The main idea is not to use any sensor in order to check whether an error has occurred, and not to conduct re-computation to address the error. Instead, we employ a *first-try-right* scheme, which always guarantees the generation of the desired *CF* independently of possible errors. To this end, we exploit the fact that particular combinations of split-errors cancel out the unwanted effects.

More precisely, how a volumetric split-errors affects the generation of the desired target-*CF* depends on two factors, namely i) the number of occurrences of such volumetric split-errors and ii) the volume of the selected droplets (smaller or larger one) in such an erroneous mix-split step. Note that we have already discussed the effect of the first factor (i) to the target-*CF* in Example 2. To this end, we need to introduce the term *error-vector* first.

1) Error-vector (E): Let us assume that k volumetric split-errors may occur, i.e., in k mix-split operations, two droplets with different volumes will result. Then, an error-vector $E = [+, -]^k$ denotes whether the smaller (denoted by $-$) or the larger (denoted by $+$) daughter-droplet is selected following these split-steps. More precisely, an error vector $[+, -, +, -]$ corresponding to the mix-split operations $\{O_1, O_2, O_3, O_4\}$ denotes that volumetric split-errors have occurred in these four steps, and following the first and the third mix-split operations, the daughter-droplet with the larger volume is used for further execution of the assay while after errors at the second and the fourth mix-split operations, the one with the smaller volume is taken. Note that one of the two daughter-droplets is randomly chosen (while the other is discarded) in the classical *twoWayMix* dilution algorithm [8]. Since k errors may occur, 3^k possible error-vectors are possible. Therefore, it is interesting to study the properties of an error vector that maximizes the error in target-*CF*.

Example 3. *Consider the mix-split graph/tree shown in Fig. 1 which, ideally yields a droplet with CF =* $\frac{25}{32}$*. Furthermore, assume that a volumetric-split error occurs at the third and fourth mix-split step of this mix-split tree. Fig. 4 demonstrates the effect of split-errors on the target-CF.*

We observe that some subsets of errors often cancel out the effects of single errors and, hence, do not have an effect to the target-*CF* (in the following, we call such a set a *non-critical set of errors*). On the other hand, some subsets of error may affect the target-*CF* badly (critical errors), for which we need to modify the corresponding mix-split steps using a roll-forward approach [18]. As a result, the effect of all possible errors cancel out each other, and ultimately, the desired target-*CF* (within the error-tolerance limit) is achieved. The following example illustrates the ideas.

TABLE I
COMPARATIVE FEATURES OF THE PROPOSED WORK AGAINST PRIOR ART

Method	Cyber-physical based?	Recovery mechanism	Recovery guaranteed?[1]	Consider worst-case error scenario?[2]	Reliable for multiple errors?	On-chip sensing needed?	Perform Re-synthesis?	Steps needed for recovery?
Zhao *et al.* [22]	yes[5]	rollback[3]	no	no	no	yes	yes	indefinite
Alistar *et al.* [26]	yes[5]	re-merge and re-split[4]	no	no	no	yes	yes	indefinite
Alistar *et al.* [27]	yes[5]	rollback[3]	no	no	no	yes	yes	indefinite
Luo *et al.* [23]	yes[5]	rollback	no	no	no	yes	yes	indefinite
Luo *et al.* [24]	yes[5]	rollback	no	no	no	yes	yes	indefinite
Hsieh *et al.* [25]	yes[5]	rollback (dynamic)	no	no	no	yes	yes	indefinite
Poddar *et al.* [18]	yes[5]	roll-forward	yes	no	yes	yes	yes	definite
Li *et al.* [36]	yes[6]	rollback, re-merge and re-split[4]	no	no	no	yes	yes	indefinite
Li *et al.* [37]	yes[6]	rollback and droplet-volume regulation	yes	no	yes	yes	yes	indefinite
Proposed	no[5]	roll-forward	yes	yes	yes	no	no	definite

[1] split-error might reoccur during the re-execution phase; [2] based on proper selection of erroneous daughter-droplets (larger/smaller) in a sequence of mix-split steps; [3] the recovery sub-graph is assumed to be error-free during re-execution; [4] performs re-merge and re-split operations a fixed number of times; [5] *DMFB* based; [6] *MEDA* based.

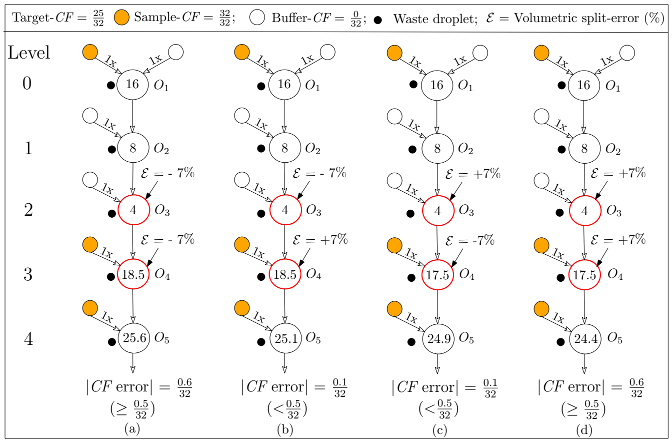

Target-$CF = \frac{25}{32}$; ● Sample-$CF = \frac{32}{32}$; ○ Buffer-$CF = \frac{0}{32}$; ● Waste droplet; $\mathcal{E}$ = Volumetric split-error (%)
Level
0
1
2
3
4
1x
16
8
4
18.5
17.5
25.6
25.1
24.9
24.4
O_1
O_2
O_3
O_4
O_5
$\mathcal{E}$ = - 7%
$\mathcal{E}$ = +7%
$\mathcal{E}$ = -7%
$|CF \text{ error}| = \frac{0.6}{32}$ $(\geq \frac{0.5}{32})$
$|CF \text{ error}| = \frac{0.1}{32}$ $(< \frac{0.5}{32})$
$|CF \text{ error}| = \frac{0.1}{32}$ $(< \frac{0.5}{32})$
$|CF \text{ error}| = \frac{0.6}{32}$ $(\geq \frac{0.5}{32})$
(a)
(b)
(c)
(d)

Fig. 4. Effects of different types of error-vector to the target-*CF*=$\frac{25}{32}$ for accuracy level 5, involving two erroneous steps.

Example 4. *Figs. 5 (a) and (b) show the effect of the error-vectors* $[+,-,-,+]$, $[+,+,+,+]$ *to the target-CF=*$\frac{25}{32}$ *for accuracy level 5. Out of* $3^4 - 1 = 80$ *possible error combinations, it is observed by exhaustive simulation that the error in the target-CF becomes maximum for the error-vector.* $[+,+,+,+]$. *However, if we consider single split-errors with the same sign, then each individual step* $\{\{O_1\}, \{O_2\}, \{O_3\}, \{O_4\}\}$ *becomes non-critical. On the other hand, by simulating all 80 errors, we observe that in the presence of all multiple split-errors,* $\{O_1, O_2, O_4\}$ *turns out to be the maximum-size non-critical subset of errors for the target-CF=*$\frac{25}{32}$ *. That is, the effect of these mix-split errors to the target-CF cancels out each other – independently of whether an error occurs or not.*

Based on this fact, the original mix-split tree *is changed only at the output of Step* O_3 *as shown in Fig. 5(c). Conducting the mix-split steps according to this graph eventually yields the desired target-CF=*$\frac{25}{32}$ *(within the error-tolerance limit) even in the occurrence of errors (and without any sensing, re-computations, etc; i.e., at first-try-right). Note that in the modified graph, error occurring in Step* O_3*, if any, will cancel out at Step* O_4*, because both the two erroneous droplets (one larger and one smaller), are allowed to execute identical actions before merger at* O_4 *[18].*

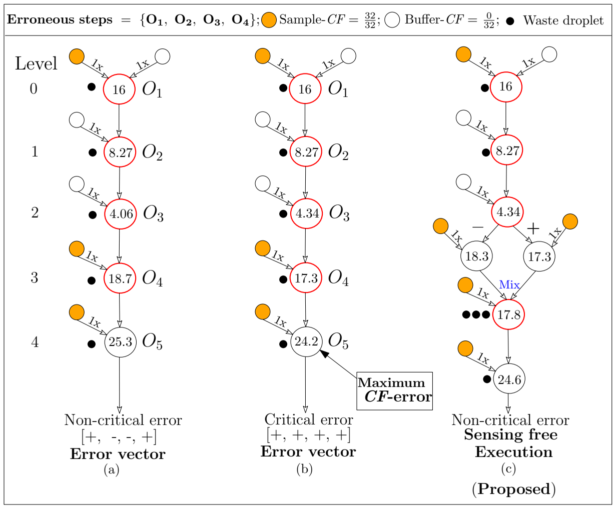


Fig. 5. General idea of the proposed solution.

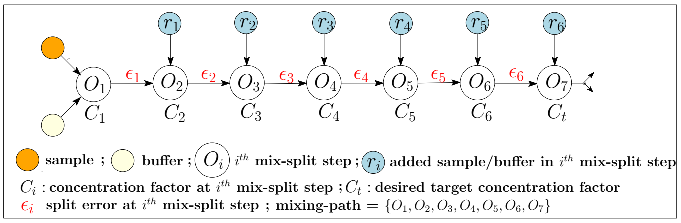
IV. Effect of Errors

In this section, we analyze the impact of volumetric split-errors on the target-*CF* during sample preparation.

A. Effect of Multiple Volumetric Split Errors

It has been experimentally observed that a split-error caused by a volumetric imbalance may affect the desired *CF* [18]. Let ϵ_i indicate the percentage of the volumetric split-error occurring at the i^{th} mix-split step. A fundamental question in this context is the following: *"How is the* CF *of a target-droplet affected by multiple volumetric split-errors* $\{\epsilon_1, \epsilon_2, \ldots, \epsilon_{i-1}\}$ *occurring on different mix-split steps in the mixing path during sample preparation?"*.

In order to find a reasonable answer to the above question, let us consider the dilution problem for generating a target *CF* = C_t using *twoWayMix* [8] as shown in Fig. 6, where O_i represents the i^{th} (1:1) mix-split step, C_i is the resulting *CF* after the i^{th} mix-split step, and r_i is the *CF* of the sample (100%)/buffer (0%) used in i^{th} mix-split operation. Without loss of generality, let us assume that a volumetric split-error ϵ_i occurs after the i^{th} mix-split step of the mixing path, i.e., after a split of 2X-droplet, the volume of two daughter-droplets becomes 1+ϵ and 1-ϵ. Initially, sample and buffer are mixed at the first mix-split step (O_1). After this mixing operation, the *CF* and volume of the resulting droplet becomes $C_1 = \frac{1}{2}$ and $V_1 = 1$, respectively, because $C_1 = \frac{P_0 \times (1 \pm \epsilon_0) + 2^{-1} \times r_0}{Q_0 \times (1 \pm \epsilon_0) + 2^{-1}}$ and $V_1 = \frac{Q_0 \times (1 \pm \epsilon_0) + 2^{-1}}{2^0}$, where $P_0 = Q_0 = \frac{1}{2}$, $\epsilon_0 = r_0 = 0$. Note that $r_i = 1$ (0) indicates whether a sample (buffer) is used in the i^{th} mix-split step of the mixing path. Furthermore, the sign $+$ ($-$) in the expression indicates whether a larger (smaller) droplet is used in the next mix-split step followed by a split operation.



r_1
r_2
r_3
r_4
r_5
r_6
O_1
O_2
O_3
O_4
O_5
O_6
O_7
ϵ_1
ϵ_2
ϵ_3
ϵ_4
ϵ_5
ϵ_6
C_1
C_2
C_3
C_4
C_5
C_6
C_t
sample ;
buffer ;
O_i i^{th} mix-split step ;
r_i added sample/buffer in i^{th} mix-split step
C_i : concentration factor at i^{th} mix-split step ; C_t : desired target concentration factor
ϵ_i split error at i^{th} mix-split step ; mixing-path = $\{O_1, O_2, O_3, O_4, O_5, O_6, O_7\}$

Fig. 6. Mix-split operations for generating target-*CF* = C_t with accuracy level n = 7.

As a result of a volumetric split-error at step S_1, the daughter-droplets may differ in volume by the factor $1 \pm \epsilon_1$ which, in turn, affects the concentration of the produced droplets. As a result of this volumetric imbalance, the concentration and volume of the generated droplets after the mix-split operation S_2 will become $C_2 = \frac{P_1 \times (1 \pm \epsilon_1) + r_1}{Q_1 \times (1 \pm \epsilon_1) + 2^0}$ and $V_2 = \frac{Q_1 \times (1 \pm \epsilon_1) + 2^0}{2}$. Note that, here, $P_1 = P_0 \times (1 \pm \epsilon_0) + 2^{-1} \times r_0$ and $Q_1 = Q_0 \times (1 \pm \epsilon_0) + 2^{-1}$. Moreover, due to another volumetric split-error (ϵ_2) in the next mix-split operation, the concentration and volume of the generated droplets after the mix-split operation O_3 becomes $C_3 = \frac{P_2 \times (1 \pm \epsilon_2) + 2 \times r_2}{Q_2 \times (1 \pm \epsilon_2) + 2}$ and $V_3 = \frac{Q_2 \times (1 \pm \epsilon_2) + 2}{2^2}$, respectively, where $P_2 = P_1 \times (1 \pm \epsilon_1) + r_1$ and $Q_2 = Q_1 \times (1 \pm \epsilon_1) + 2^0$. Similarly, if an unbalanced volumetric split-error (ϵ_3) occurs at O_3, concentration and volume of the produced droplets of O_4 becomes $C_4 = \frac{P_3 \times (1 \pm \epsilon_3) + 2^2 \times r_3}{Q_3 \times (1 \pm \epsilon_3) + 2^2}$ and $V_4 = \frac{Q_3 \times (1 \pm \epsilon_3) + 2^2}{2^3}$, where $P_3 = P_2 \times (1 \pm \epsilon_2) + 2 \times r_2$ and $Q_3 = Q_2 \times (1 \pm \epsilon_2) + 2$. As an outcome of these split-errors, occurring after each mix-split operation, concentration and volume of the target-droplets at the end of the mix-split operation (O_7) becomes $C_7 = \frac{P_6 \times (1 \pm \epsilon_6) + 2^5 \times r_6}{Q_6 \times (1 \pm \epsilon_6) + 2^5}$ and $V_7 = \frac{Q_6 \times (1 \pm \epsilon_6) + 2^5}{2^6}$, where $P_6 = P_5 \times (1 \pm \epsilon_5) + 2^4 \times r_5$ and $Q_6 = Q_5 \times (1 \pm \epsilon_5) + 2^4$.

TABLE II
Effect of volumetric split-errors to the target-*CF* = $\frac{25}{128}$ for accuracy level $n = 7$.

Erroneous steps*	*Produced* CF×128	CF *error (%)*	Non-critical set?
$\{O_1\}$	25.04	0.17%	Yes
$\{O_1, O_3\}$	25.14	0.02%	Yes
$\{O_1, O_2, O_3\}$	25.02	0.08%	Yes
$\{O_1, O_2, O_3, O_4\}$	25.43	1.72%	Yes
$\{O_1, O_2, O_3, O_4, O_6\}$	26.28	5.11%	No
$\{O_1, O_2, O_3, O_4, O_5, O_6\}$	27.59	10.36%	No

* for error-vector $\{O_1, O_2, O_3, O_4, O_5, O_6\} \rightarrow [+,+,+,+,+,+]$.

Hence, for the occurrence of multiple volumetric split-errors, say $\{\epsilon_1, \epsilon_2, \epsilon_3, \ldots, \epsilon_{i-1}, \epsilon_{i-1}\}$ at different mix-split steps of the mixing path, the *CF* and volume of the generated target-droplet after the final mix-split operation can be computed using the following expressions:

$$C_i = \frac{P_{i-1} \times (1 \pm \epsilon_{i-1}) + 2^{i-2} \times r_{i-1}}{Q_{i-1} \times (1 \pm \epsilon_{i-1}) + 2^{i-2}} \tag{1}$$

$$V_i = \frac{Q_{i-1} \times (1 \pm \epsilon_{i-1}) + 2^{i-2}}{2^{i-1}} \tag{2}$$

where $P_i = P_{i-1} \times (1 \pm \epsilon_{i-1}) + 2^{i-2} \times r_{i-1}$ and $Q_i = Q_{i-1} \times (1 \pm \epsilon_{i-1}) + 2^{i-2}$. In this way, the impact of multiple volumetric split-errors occurring on different mix-split steps of the mixing path on the target-*CF* can be precomputed.

B. Critical and Non-critical Set of Errors

Note that an attempt was made in [18] for measuring the role of a volumetric split-error to the target-*CF*. However, only single volumetric split-error was considered in that model. In this work, we consider *multiple volumetric split-errors* (which can appear in any combination on the *mix-split graph*). We measure their effects on the target-*CF* using Expression (1) and Expression (2) and based on that, we classify them as being *critical* or *non-critical*. A set of multiple volumetric split-error belongs to *non-critical (critical)* set if the error in the target-*CF* induced by these set of split-errors is less than (greater or equal to) the inherent error-tolerance limit τ ($=\frac{0.5}{2^n}$) used to represent the *CF*-value in binary. Clearly, the non-critical set of errors can easily be ignored as there will be no significant change in the output-*CF*. Otherwise, corrective measures are required to restore the desired target-*CF* for the critical set of errors.

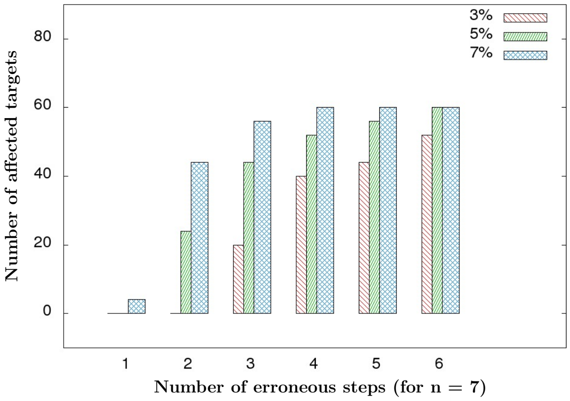


Fig. 7. Number of affected targets for different split-errors (with various % of split-errors (ϵ)) for all odd target-*CFs* with accuracy level n = 7.

TABLE III
EFFECT OF VARIOUS ERROR-VECTOR TO THE TARGET-*CF* = $\frac{25}{128}$ FOR ACCURACY $n = 7$

Serial no.	*Error vector*	*Produced* CF×128	*CF* error (%)
1	$[+,+,+,+,+,+]$	27.59	10.36%
2	$[+,+,+,+,+,-]$	25.78	7.12%
3	$[+,+,+,+,-,+]$	24.91	4.36%
4	$[+,+,+,+,-,-]$	23.22	7.12%
.	.	.	
.	$[+,+,-,+,+,+]$	27.70	10.80%
.	$[-,-,+,-,-,-]$	22.31	10.76%
.	.	.	
61	$[-,-,-,-,+,+]$	26.73	6.92%
62	$[-,-,-,-,+,-]$	24.92	0.32%
63	$[-,-,-,-,-,+]$	24.05	3.80%
64	$[-,-,-,-,-,-]$	22.37	10.52%

Example 5. *Consider multiple volumetric split errors in a mix-split graph which aims for generating a target-CF =* $\frac{25}{128}$ *and an accuracy level* n *= 7. Note that there exist* $\{3^{n'-1} - 1\}$ *possible sets of multiple split-errors of a mix-split graph containing* n' *mix-split operations. Table II demonstrates the effect of volumetric split-errors to the target-CF for some representative test cases with 7% volumetric split-errors.*

We also carried out an experiment for counting the number of affected target-*CF*s for accuracy level = 7, and report the results as barchart in Fig. 7. We conduct the same experiment for different percentages of volumetric split-errors (ϵ) = 3%, 5% and 7%. However, the population of erroneous target-*CF*s increases gradually with the multiplicity of volumetric split-errors and error-percentage.

C. Effect of various error-vectors on target-CFs

In general, an ϵ volumetric split-error produces two unequal droplets of volume $(1+\epsilon)$ and $(1-\epsilon)$. In the case of a mix-split operation involving a split-error, it is very hard to predict what droplet (the smaller or the larger one) is forwarded as reactant-droplet to the next mix-split operation in the absence of error-detection mechanism. Since the *mix-split graph* contains a number of mix-split operations, it becomes more complex when several mix-split operations suffer from split-error. It is also interesting to determine the set of split-operations that yields the maximum error at the target-*CF*. The corresponding error-vector is likely to contain a large number of critical-steps in the mix-split sequence.

Example 6. *Consider again the generation of the target-CF=*$\frac{25}{128}$ *with accuracy level 7. Let's assume that split-errors appear in each split operation of the mix-split graph (except the final one which would not alter the target-CF anyway). For the error-vector* $[+,+,-,+,+,+]$*, the error in the target-CF becomes (*$\frac{27.70}{128}$*) becomes maximum (see Table III). Searching for the vector requires* $O(N)$ *time, where* N *is number of entries in the search table. Since* $N = 3^{n'-1} - 1$*, the search time increases exponentially with* n'*.*

Note the error vector that causes maximum-error in the target-*CF* represents a worst-case scenario. However, for many other error-vectors, the deviation in the target-*CF* may also be far outside the acceptable limit. Hence, for achieving fault-tolerance, the modified mix-split graph should be made resilient to each and every such potential error-vector. Later in this section, we will describe a procedure for graph-modification that starts from an error-vector, which involves a large number of critical steps for the target-*CF*. We may start from the maximum-error-vector obtained by exhaustive simulation. However, in order to avoid the search complexity, we present below an empirical rule-based procedure that produces an error-vector, which causes near-maximum *CF*-error at the target. To this end, one needs to scan the nodes (*CF*-values) of the mix-split graph starting from the root to the leaf node (target node). More precisely, during this computation, the following two sets are determined:

- H_{cf}: The set of all intermediate *CF*s that have a larger value than the target-*CF* and
- L_{cf} : The set of all intermediate *CF*s that have lower value than the target-*CF*.

Two cases may now occur:

Case I. Let us assume that the target-*CF* lies within the interval $[\frac{1}{2^n}, \frac{2^{n-1}-1}{2^n}]$, and $\{C_i, C_j, C_k\}$ are the intermediate *CFs*, $C_i \in H_{cf}$ and $C_j \in L_{cf}$ i.e., $C_i > C_t$ and $C_j < C_t$. Now, a larger-volume droplet is selected in an erroneous node C_i (*CF* = C_i) for the edge representing mix-split operations from C_i to C_k in the *mix-split graph*, i.e., ($C_i \rightarrow C_k$). However, a smaller-volume droplet is selected at an erroneous node C_j for the edge $C_j \rightarrow C_k$.

Case II. Now, when the target-*CF* $\in [\frac{2^{n-1}+1}{2^n}, \frac{2^n-1}{2^n}]$ with $C_i \in H_{cf}$ and $C_j \in L_{cf}$, the larger-droplet is selected in an erroneous node C_j during the mix-split operation ($C_j \rightarrow C_k$), whereas the smaller-droplet is selected for the mix-split operations ($C_i \rightarrow C_k$).

We validate the rule stated above by performing a number of experiments. We run our experiments for each of the target-*CF*s with various accuracy levels, and generate the error-vector. Then, during execution, we inject volumetric split-errors at each mix-split operation of the *mix-split graph* according to the error-vector produced by the Empirical Rule. Exhaustive simulation shows that the rule indeed produces the maximum *CF*-error barring only a few examples.

Fig. 8 shows the selection of the error-vector $[+,+,-,+,+,+]$ for the target-*CF*=$\frac{25}{128}$ for accuracy level 7. The *CF*-values highlighted with red (black) indicates the deviated (expected) values due to the presence (absence) of errors in the mix-split operations. The effect of the selected erroneous droplet (+/−) for each mix-split step of the mix-split path is also indicated with a red-colored value.

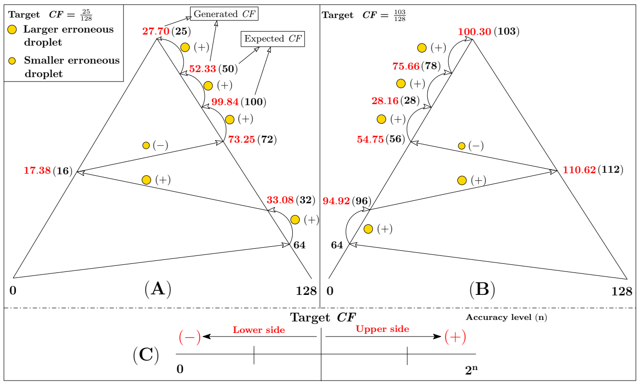


Fig. 8. Selection of erroneous droplets for target-*CFs* $\frac{25}{128}$ and $\frac{103}{128}$ with accuracy level n = 7.

We further performed experiments for determining the critical and non-critical set of errors by assuming the fact that the error occurs on different mix-split steps according to the resulting error-vector. It has been noticed that non-critical operations $\{O_1, O_2, O_3, O_4\}$ (Table II) becomes critical for the error-vector $[+,+,-,+,+,+]$.

We have performed rigorous theoretical analyses and further experiments to study the properties of maximum *CF*-error in a target-*CF*, the details of which can be found in the supplementary material [39]. Our results indicate that it may not be possible to formulate a mechanism that will identify the exact maximum-error-vector without doing exhaustive simulation. Note that given a target-*CF*, Algorithm 1, described later, selects an error-vector based on the earlier Empirical Rule.

1) Baseline approach to error-obliviousness: We present below a baseline approach for canceling the effects of multiple split-errors regardless of their criticality. The idea is to construct a complete binary mix-split tree (where each node excepting the leaves has degree two, and all leaf nodes are at depth n, where n is the accuracy level). For simplicity, we prove the theorem for n = 4. However, the argument can be extended for any general value of n.

Consider the problem for generating a target-*CF*=C_t of accuracy level n = 4. Let $\{O_1, O_2, O_3, O_4\}$ be the sequence of (1:1) mix-split operations which needs to be performed for generating the target-*CF*. The complete dilution tree for producing 2^4 = 16 target-droplets with *CF*=C_t is shown in Fig. 9. In this complete dilution tree, all nodes present in a particular depth are identical and a node at i^{th} depth corresponds to the $(i+1)^{th}$ mix-split operation O_{i+1} for the target-*CF*=C_t. For example, nodes O_2^L and O_2^R in depth 1 are identical and they represent the second mix-split operation O_2 for the target-*CF*=C_t. The target-droplets $(d_1 - d_{16})$ appear as leaf node of the dilution tree, and in the normal condition (in the absence of volumetric errors) each of them will have the same *CF*-value. We can now prove the following result.

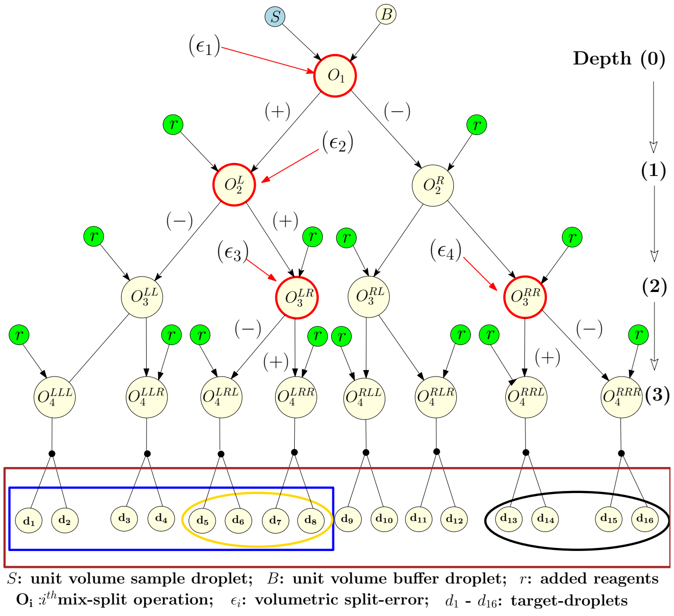


Fig. 9. Error-free sample preparation with multiple split-errors (n=4).

Theorem. If the target-droplets $\{d_1, d_2, \ldots, d_{2^n}\}$ produced by the complete dilution tree (Fig. 9) are mixed together at the end, without discarding any intermediate droplets, then the *CF* of the resulting mixture becomes exactly C_t even if volumetric split-errors occur in the mixing tree.

Proof. Without loss of generality, let us assume that while executing the operations of the dilution tree (Fig. 9), mix-split operations O_1, O_2^L, O_3^{LR}, and O_3^{RR} (red colored nodes) are affected by the split-errors $\epsilon_1, \epsilon_2, \epsilon_3$, and ϵ_4. Note that an erroneous split operation produces two unbalanced daughter-droplets, one with a larger volume (+) and another with a smaller volume (-), as shown in Fig. 9. However, the effect of an erroneous split-operation to the target-*CF* is canceled-out by mixing the target-droplets (produced by the two erroneous droplets) together at the end, when the error occurs at their nearest-common-ancestor mix-split node (step) [18]. For example, if we mix target-droplets (d_5, d_6) and (d_7, d_8) together at the end, the effect of splitting-error ϵ_3 (occurred at the nearest-common-ancestor mix-split node O_3^{LR}) is canceled-out. Although mixing of (d_5, d_6) and (d_7, d_8) removes the effect of ϵ_3, it does not remove the effect of ϵ_2. However, the effect of ϵ_2 on the target-droplet is canceled when droplets (d_1, d_2, d_3, d_4) and (d_5, d_6, d_7, d_8) are mixed together at the end. Thus, the overall effects of ϵ_2 and ϵ_3 on target-droplets d_1 to d_8 are removed when they are mixed. Similarly, we can argue that the effect of ϵ_4 is corrected at the end by mixing target-droplets (d_{13}, d_{14}) and (d_{15}, d_{16}). Finally, mixing of target-droplets (d_1-d_8) and (d_9-d_{16}) not only removes the effect of ϵ_1 split-error but also cancels the effects of ϵ_1, ϵ_2 and ϵ_3 at the end. Hence, mixing of all target-droplets together produces a mixture with *CF* = C_t regardless of the number of volumetric split-errors in the mixing tree. This argument can be generalized for any complete dilution tree. ■

Therefore, it is possible to generate any target-*CF* correctly from the *complete dilution tree* regardless of the number of volumetric split-errors. However, this baseline approach increases sample-preparation time and reactant-cost significantly, as the dilution tree contains 2^n-1 mix-split nodes.

In order to remove the above shortcomings, we create a small-size[1] graph (called *error-tolerant graph*) for producing the target-*CF* within the error-tolerance limit. We may start selecting an arbitrary error-vector (E), and then prune/modify the complete binary tree by mapping each critical and non-critical error in E to the corresponding node of the complete dilution tree for the given target-*CF*. However, it is preferable to start from an error-vector that causes the maximum-error in target-*CF* (obtained by exhaustive search) or from the one produced by the Empirical Rule, which produces an error-vector that "almost" maximizes the error in the target-*CF*.

V. Resulting Methodology

The working principle of the proposed method for generating the target-*CF* = C_t within the user specific error-tolerance bound (τ) on a digital microfluidic platform is as follows.

- Construct the mix-split path for generating the target-*CF* = C_t using the basic *twoWayMix* algorithm [8] and the corresponding complete dilution tree (T) [38].
- Depending on the maximum allowable split error (ϵ), determine the error-vector by the Empirical Rule as described in Section IV-C;
- Determine the initial set of "critical-steps" (considering only one split-error at a time) with the sign matching with that of the error-vector obtained in the previous step. Let nc_{steps} denote the set of remaining steps, which are individually non-critical for this error-vector.
- Now we need to check the criticality of all mix-split steps belonging to nc_{steps} from a collective viewpoint (i.e., in the presence of simultaneous occurrences of two or more such errors). In order to test this, we consider all $3^{|nc_{steps}|-1}-1$ combinations (+, -, ϕ (no-error)) of error-vectors. A k-size subset of nc_{steps} is said to be *finally non-critical* if for each assignment of + and - on these k-positions, the corresponding multiple split-error, when inserted, causes the target-*CF* to lie within the allowable limit. We begin the search starting from higher-to-lower order (i.e., first with the combinations that comprises no ϕ, and later increasing their occurrences), and exit as soon as a non-critical subset of steps is identified. This will be a maximal-size non-critical subset of steps (nc_{max}) on the reaction-path. The remaining steps belonging to $\{nc_{steps} \setminus nc_{max}\}$ are marked critical, and are added to the initial set of critical steps.
- For each mix-split operation at i^{th} depth of T, starting from the root (depth 0).
  - If $i \notin nc_{max}$
    - ∗ Retain all the nodes of depth (i+1), which have parent-child relationship with the existing nodes of depth i.
  - If $i \in nc_{max}$
    - ∗ Keep a single node at depth (i+1) of T and delete all other nodes.[2]

We present below two algorithms: Algorithm 1 generates an error vector based on the empirical rule discussed earlier in Section IV-C; Algorithm 2 returns a maximal-size non-critical subset of mix-split steps based on the working principle discussed in this section.

Algorithm 1: Error-vector based on empirical rule.

```
Input: Target-CF=$C_t$, accuracy label $n$, size of the mixing path $l$, maximum
       allowable split-error $\epsilon$.
Output: Error vector $E_v$ that causes a near-maximum error in the target-CF=$C_t$.
1  Let $I_{cfs}$ = $[Cf_1, Cf_2, Cf_3, \ldots, Cf_l]$ be the CFs corresponding to the
   sequence of intermediate mix-split operations for the target-CF;
2  Let $E_v$ = [] // used for storing the error-vector;
3  for $(i = 0; i < |l|; i = i + 1)$ do
4      //for each mix-split step;
5      if $C_t \in [\frac{1}{2^n}, \frac{2^{n-1}-1}{2^n}]$ then
6          if $I_{cfs}[i] > C_t$ then
7              $E_v.append(+\epsilon)$
8          else
9              $E_v.append(-\epsilon)$
10     if $C_t \in [\frac{2^{n-1}+1}{2^n}, \frac{2^n-1}{2^n}]$ then
11         if $I_{cfs}[i] > C_t$ then
12             $E_v.append(-\epsilon)$
13         else
14             $E_v.append(+\epsilon)$
15 return $E_v$
```

*Algorithm 3(Line 3-5) takes care of the case when C_t becomes $\frac{2^{n-1}}{2^n}$.

Given the desired target-*CF* = C_t, accuracy label (n), the required number of mix-split steps (l), and the maximum allowable volumetric split-error, Algorithm 1 calculates *CF* of the intermediate fluid following each mix-split operation (Line 1) and based on that it determines the corresponding error-vector (Lines 3-14).

Example 7. *Consider target-CF = $\frac{25}{128}$ for accuracy level 7 and allowable split-error 7%. At first, it calculates the CF-value after each mix-split operation while generating the target-CF: I_{cfs} = {64, 32, 16, 72, 100, 50}, using the twoWayMix method. Next, by scanning the intermediate CFs (I_{cfs}) from left-to-right, Algorithm 1 returns the error-vector [+, +, -, +, +, +] for the target-CF [see Fig. 8].*

Algorithm 2 returns a maximal-size non-critical set of errors depending on the target-*CF* C_t, accuracy n, the error-vector returned by the Algorithm 1, and the error-tolerance limit τ. Procedure Gen_{CF} returns the generated *CF*-value based on the positions of split-errors and the error-signs. For example, if we insert non-zero error in all six mix-split steps (excepting the last one), there will be 63 possible error-vectors for the target-*CF* $\frac{25}{128}$ for accuracy level = 7. The generated *CF*-errors

[1] reduced # of mix-split steps

[2] Note that during bioasssay execution, all droplets that are generated at the i^{th} mix-split step are mixed together from which a 1X-size droplet is extracted for subsequent use.

Algorithm 2: Find maximal non-critical (nc_{max}) steps.

```
   Input: Target-CF=$C_t$, accuracy label $n$, error-vector $e_v$ and error tolerance limit
          = $\tau$.
   Output: Maximal-size non-critical set of errors ($nc_{max}$) =
          $\{O_1, O_2, O_3, \ldots O_k\}$, where $1 \leq k \leq n$.
1  Express $C_t$ as $n$-bit binary number ($bin_{C_t}$) $\frac{x}{2^n}$;
2  Remove the low-order zero bits from $bin_{C_t}$;
3  Let $l$ = length($bin_{C_t}$) - 1;
4  Let $S$ = [1, . . . , $l$];
5  Let $NC_{steps}$ = [] // used for storing all possible non-critical steps;
6  Let $nc_{max}$ = [] // used for storing max-length non-critical steps;
7  for $(i = 0; i < |S|; i = i + 1)$ do
8      //for each position of $S$;
9      $C = Gen_{CF}(S[i], e_v[i], l, bin_{C_t})$
10     if $\frac{|C_t - C|}{2^n} < \tau$ then
11         $NC_{steps}.append(S[i])$ // $S[i]$ is merged as non-critical step
12 Let $S$ be the all possible subsets of $nc_{steps}$, except the null set; // for finding
   the remaining critical-steps w.r.t other error-vectors;
13 for $(i = |S| - 1; i >= 0; i = i - 1)$ do
14     //for each subset of $S$;
15     Let $E_v$ = all error-vecotrs of length $S[i]$;
16     flag = 0;
17     for $(j = 0; j < |E_v|; j = j + 1)$ do
18         //for each error-vector of $E_v$;
19         $C = Gen_{CF}(S[i], E_v[j], l, bin_{C_t})$;
20         if $\frac{|C_t - C|}{2^n} >= \tau$ then
21             flag = 1;
22             break;
23     if flag == 0 then
24         $nc_{max}$ = $S[i]$ // max-length non-critical set obatined;
25         break;
26 return $nc_{max}$
1  Procedure Gen$_{CF}$ ($E_p, E_v, l, bin_{C_t}$)
3      $P_1 = \frac{1}{2}, Q_1 = 1, C_1 = \frac{1}{2}, V_1 = 1$;
4      for $(j = 2; j <= l + 1; j = j + 1)$ do
5          $\epsilon$ = 0;
6          if $(j - 1) \in E_p$ then
7              $\epsilon$ = $E_v[E_p.index(j - 1)]$
8          end
9          $P_j = P_{j-1} \times (1 + \epsilon) + 2^{j-2} \times bin_{C_t}[j]$;
10         $Q_j = Q_{j-1} \times (1 + \epsilon) + 2^{j-2}$;
11         $C_j = \frac{P_j}{Q_j}, V_j = \frac{Q_j}{2^{j-1}}$;
12         /* The subscript of $P_j$, $Q_j$ and $C_j$ changes according to the value of
           the for-loop j and $P_1$, $Q_1$ and $C_1$ are initialized in line 3;
13     end
15     return $C_j$
```

corresponding to these error-vectors are shown along Y-axis in Fig. 10 for split-error ϵ = .07. In the X-axis, the error-vectors are sequenced following a gray-code (assuming + →0, and − →1). Note that the error in the target-*CF* exceeds the safe limits for a number of error-vectors. Algorithm 2 returns $\{O_1, O_2, O_3\}$ as the maximal-size non-critical steps. Other non-critical maximal sets also exist such as $\{O_1, O_2, O_4\}$ corresponding to the error-vector [+, +, -, +, +, +]. Finally, Algorithm 3 creates the error-tolerant graph by pruning and modifying the complete binary mix-split tree based on the non-critical set of errors (nc_{max}) produced by Algorithm 2.

It is easy to prove that the modified mix-split graph constructed by Algorithm 3 will tolerate multiple split-errors of any sign. Needless to say, if an error occurs at the critical nodes, they are canceled because of the roll-forward mechanism inserted there. For the remaining non-critical nodes, no error-vector can produce a critical *CF*-error at the target because all of them have been exhaustively checked while determining the non-critical set of steps. An example of creating the error-tolerant dilution graph can be found in the supplementary file [39]. The produced *CF* for all possible error vectors for the target-*CF* = $\frac{25}{128}$ is shown in Fig. 11. Note that for all combinations of split-errors inserted in the graph, the resulting target-*CF* will always lie within the allowable limits $\pm\frac{0.5}{128}$.

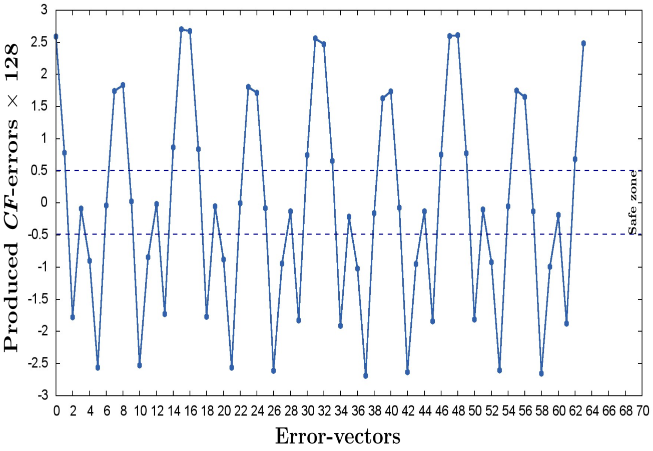


Fig. 10. Generated *CF*-values corresponding to various error-vectors.

Algorithm 3: Error-tolerant dilution tree (T).

```
   Input: Target-CF=$C_t$, maximum allowable split-error $\epsilon$, error-tolerance limit $\tau$,
          accuracy label $n$.
   Output: Error-tolerant dilution ($T$) tree for the target-CF=$C_t$.
1  Let $O$ = $[O_1, O_2, O_3, \ldots, O_k]$ be the set of mix-split operations need to be
   performed for generating the target-CF=$C_t$, $1 \leq k \leq n$;
2  Let $T$ be the complete dilution tree [38];
3  if length(O)==1 then
4      //execute when target-CF = $\frac{2^{n-1}}{2^n}$;
5      return $T$
6  else
7      //for other CFs;
8      Let $E_v$ be the error-vector returned by the Algorithm 1 ;
9      Let $nc_{max}$ be the maximal-size non-critical set of error returned by the
       Algorithm 2 ;
10     Let $CR_{ops}$ =$[c_1, c_2, c_3, \ldots, c_q]$ be the critical set of errors, $0 \leq q \leq k$ ;
11     for $(i = 0; i < |O|; i = i + 1)$ do
12         //for each depth of the dilution tree ($T$), starting from 0;
13         if $i \in nc_{max}$ then
14             Keep a signle node at depth ($i$+1) of $T$ and delete all other nodes.
15         else
16             Retain all the nodes at depth ($i$+1) in $T$, which have parent-child
               relationship with the existing nodes at depth $i$.
17     return $T$
```

Handling residue-leaving (imperfect) splitting: As illustrated earlier in Fig. 2, some split operations may leave a small-size residual droplet in addition to producing two daughter-droplets. Without using any sensors, such errors can be handled by incurring some routing overhead: after each splitting, one of the daughter droplets should be shifted away from the split-location and the other one is moved towards the cell where the split-operation was performed. As a result, the residue will be merged with latter droplet, and hence, finally, only two daughter-droplets will remain. Any volumetric imbalance, if present, can be handled using the proposed method as discussed before in this section.

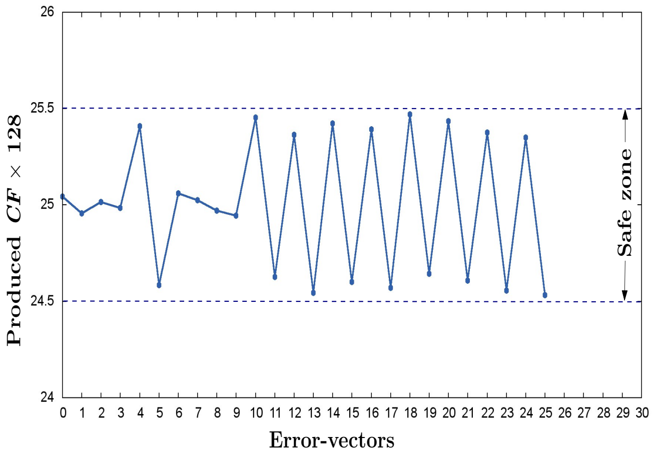


Fig. 11. *CF*-values generated by the error-tolerant graph in the presence of multiple split-errors.

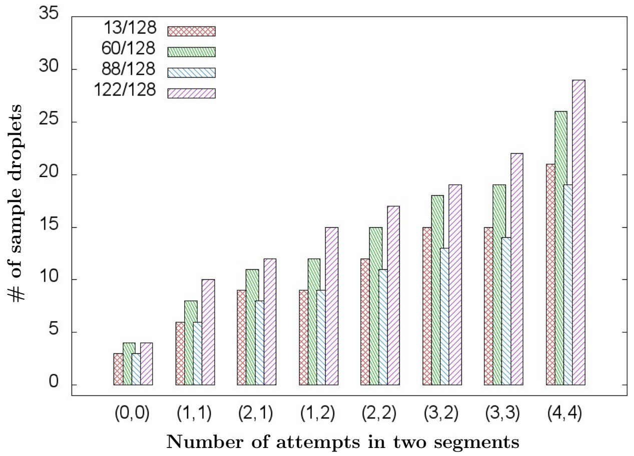


Fig. 12. Consumption of sample droplets for preparing different target-*CFs* with accuracy level 7 by the rollback procedure.

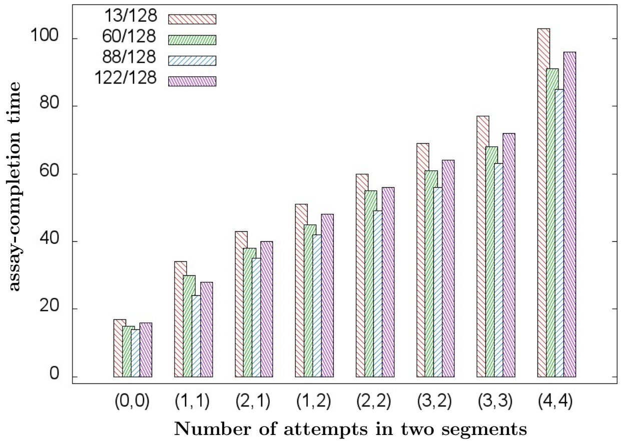


Fig. 13. Generation time for different target-*CF*s with accuracy level n = 7 by the rollback procedure.

VI. Experimental Results

We have implemented the proposed sample-preparation algorithm in Python and used a computing platform with 2 GHz Intel Core i5 processor, 8 GB memory, and 64-bit Ubuntu 14.04 operating system. We compare our results with a baseline approach and conventional rollback error-recovery approaches [22], [23].

For evaluating the performance of a rollback procedure, we insert two checkpoints (one near the middle and another at the end) in the dilution tree of each target-*CF*. Thus, the mixing-path is divided into two segments. If an on-chip sensor detects an error at the end of a segment, then operations are re-executed starting from the initial position of that segment. We performed rigorous experiments for some representative target-*CF*s and observed the cost and time overhead by assuming that two 1X-volume droplets are mixed in three units of time, and sensing including transportation requires five units of time. We observe that reactant-cost and assay-completion time increase significantly when the number of recovery-attempts per segment increases as shown in Fig. 12 and Fig. 13, respectively.

We perform various experiments on different real-life test-cases [40] as well as synthetic test-cases for evaluating performance of the baseline approach, the rollback-based error-recovery, and the proposed method. We randomly inject a number of volumetric split-errors on different mix-split steps of the mixing graph during experiments. For the rollback approach, we assume an optimistic situation, where split-errors, once being sensed, is corrected within at most two (three) attempts in each segment, i.e., within error-rate[3] 2 (3).

We report the generated *CF*-values, the number of target-droplets (load), *CF*-error, and the error-limit for each target-*CF* of accuracy level (n) = 7 in Table IV. We iterate our experiment 20 to 50 times for each target-*CF* and report the value of the the maximum *CF*-error in Table IV. We show results only for the baseline and the proposed method as there was no information available for rollback approaches [4]. We observe that the proposed method reduces the number of target-droplets significantly compared to baseline, yet efficiently bounds the target *CF*-error within the allowable error-limit. Moreover, it can be seen from Table V that it reduces the consumed sample, buffer droplets as well operation-count compared to baseline and rollback approaches, noticeably. Additionally, the re-execution-based error-recovery approaches produce more intermediate waste droplets compared to the proposed method. We also report the completion times for some representative target-*CF*s for accuracy level = 7 for rollback, baseline, and the proposed method in Fig. 14.

We further performed additional experiments to find the minimum number of target-droplets that are to be produced (load) for error cancellation for each target-*CF*. During simulation, we injected a fixed percentage of volumetric split-

[3] the number of error-occurrences on the mix-split steps (which determines how many times the rollback mechanism is to be invoked)

[4] the target-droplets generated by the existing rollback approach may contain small amount of *CF*-error (caused by some split-induced volumetric imbalance) which is not detectable by the sensor

TABLE IV
PERFORMANCE EVALUATION OF DIFFERENT METHODS.

Real-life test-cases

CF	Baseline			Proposed			$\lvert CF - \overline{CF} \rvert < \frac{0.5}{128}$?
	$\overline{CF}$	$\lvert CF - \overline{CF} \rvert$	n_d	$\overline{CF}$	$\lvert CF - \overline{CF} \rvert$	n_d	
$\frac{13}{128}$	$\frac{13}{128}$	0	128	$\frac{12.87}{128}$	$\frac{0.13}{128}$	16	yes
$\frac{27}{128}$	$\frac{27}{128}$	0	128	$\frac{26.79}{128}$	$\frac{0.21}{128}$	16	yes
$\frac{38}{128}$	$\frac{38}{128}$	0	64	$\frac{37.89}{128}$	$\frac{0.11}{128}$	4	yes
$\frac{51}{128}$	$\frac{51}{128}$	0	128	$\frac{50.85}{128}$	$\frac{0.15}{128}$	8	yes
$\frac{122}{128}$	$\frac{122}{128}$	0	64	$\frac{122.35}{128}$	$\frac{0.35}{128}$	8	yes

Synthetic test-cases

CF	Baseline			Proposed			$\lvert CF - \overline{CF} \rvert < \frac{0.5}{128}$?
	$\overline{CF}$	$\lvert CF - \overline{CF} \rvert$	n_d	$\overline{CF}$	$\lvert CF - \overline{CF} \rvert$	n_d	
$\frac{43}{128}$	$\frac{43}{128}$	0	128	$\frac{42.63}{128}$	$\frac{0.37}{128}$	4	yes
$\frac{65}{128}$	$\frac{65}{128}$	0	128	$\frac{64.90}{128}$	$\frac{0.10}{128}$	16	yes
$\frac{77}{128}$	$\frac{77}{128}$	0	128	$\frac{76.83}{128}$	$\frac{0.17}{128}$	8	yes
$\frac{90}{128}$	$\frac{90}{128}$	0	64	$\frac{89.98}{128}$	$\frac{0.12}{128}$	4	yes
$\frac{105}{128}$	$\frac{105}{128}$	0	128	$\frac{104.61}{128}$	$\frac{0.39}{128}$	16	yes

$\overline{CF}$: generated *CF*; n_d: the number of generated droplets.

TABLE V
PERFORMANCE OF DIFFERENT METHODS.

Real-life test-cases

CF	Rollback				Baseline				Proposed			
	n_m	n_s	n_b	n_w	n_m	n_s	n_b	n_w	n_m	n_s	n_b	n_w
$\frac{13}{128}$	25(32)*	12(15)	17(22)	27(25)	127	13	115	0	18	3	16	3
$\frac{27}{128}$	27(29)	15(18)	14(16)	27(32)	127	27	101	0	18	5	14	3
$\frac{38}{128}$	19(27)	9(14)	12(18)	19(30)	63	19	45	0	8	3	6	5
$\frac{51}{128}$	21(32)	12(18)	12(19)	22(35)	127	51	77	0	11	3	9	4
$\frac{122}{128}$	21(27)	17(22)	8(10)	23(30)	63	61	3	0	10	10	2	4

Synthetic test-cases

CF	Rollback				Baseline				Proposed			
	n_m	n_s	n_b	n_w	n_m	n_s	n_b	n_w	n_m	n_s	n_b	n_w
$\frac{43}{128}$	25(32)	15(19)	14(18)	27(35)	127	43	85	0	9	4	6	6
$\frac{65}{128}$	25(32)	6(9)	22(28)	26(35)	127	63	65	0	18	9	10	3
$\frac{77}{128}$	25(32)	15(19)	14(18)	27(35)	127	77	51	0	11	7	5	4
$\frac{90}{128}$	21(27)	14(18)	11(14)	23(30)	63	13	51	0	8	4	5	5
$\frac{105}{128}$	25(32)	14(18)	15(19)	27(35)	127	105	23	0	19	14	6	4

n_m: number of (1:1) mixing; n_w: number of waste droplets; $n_s(n_b)$: number of sample (buffer) droplets; *: for error-rate = 2(3).

error on each mix-split step of the mixing path and observed that the minimum number of target-droplets required to cancel 3% of the volumetric split-error is 8, whereas for 5% and 7% split-error, it increases to 16. Also, most of the target-*CF*s (80%) require an output load of only 8 droplets for up to 7% split-errors. Note that no further mixing or splitting is required for target-droplets; only dispensing is needed from the target-reservoir. However, larger-size (4X, 8X, 16X) droplets need mixing and splitting when one or more critical steps appear consecutively on the reaction path before reaching the target-*CF*. In order to study their frequencies, we have performed further experiments for all target-*CF*s with accuracy level = 7, assuming 3%, 5%, and 7% multiple volumetric split-errors and report the results in Fig. 15.

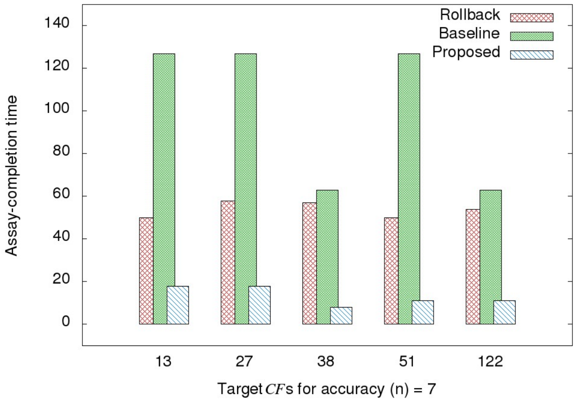


Fig. 14. Total time required to generate the desired target-*CF*s of accuracy level 7 by various methods.

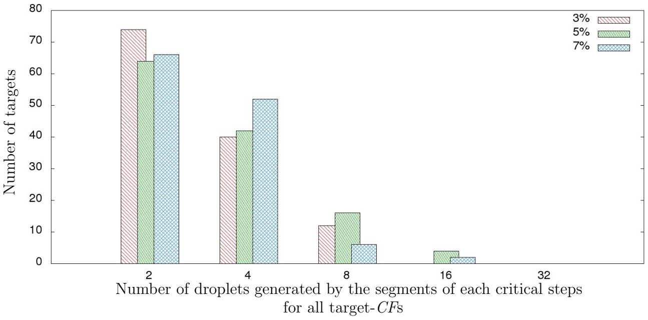


Fig. 15. Number of target-*CF*s that require reservoirs of size 4X, 8X, and 16X for storing intermediate droplets, when accuracy level = 7 and ϵ is varied.

Note that the maximum number of intermediate droplets generated by consecutive critical segments determines the largest size of the on-chip reservoir (because these droplets will be stored therein). We observe from experiments that no more than three consecutive critical steps appear for accuracy level of $CF \leq 7$, and hence, one reservoir of size 16X (or one reservoir of size 4X, 8X and 16X each, i.e., a total of three, if we want to avoid unloading of droplets) will be sufficient for storing all intermediate droplets on the chip. The droplets, after mixing operations, can simply put into the reservoir and a single droplet can then easily egress from the reservoir with just one dispense operation as needed [3], [9], [33], [41]. Dispensing a droplet from a reservoir also reduces the cumulative split-error that may otherwise creep in when a 1X droplet is produced by sequentially splitting into two halves, in each step, a large-size droplet such as 16X, 8X, or 4X.

We further report the comparison of [18] and the proposed error-oblivious method in Table VI under different error-rates. We randomly insert a number of volumetric split-errors (7%) on six potential mix-split steps of the dilution tree for five target-*CF*s. We report the required number of sample (buffer) droplets, generated waste droplets, and the total assay-completion time (mixing time + sensing time) in Table VI.

TABLE VI
PERFORMANCE EVALUATION OF [18] AND THE PROPOSED ERROR-OBLIVIOUS METHOD.

Target-*CF*	Methods	E_r = 0*				E_r = 1(2)				E_r = 3(4)			
		n_s	n_b	n_w	e_t	n_s	n_b	n_w	e_t	n_s	n_b	n_w	e_t
$\frac{27}{128}$	[18]	4	4	6	36	5(5)	6(10)	7(7)	55(77)	5(5)	12(14)	5(3)	83(89)
	Proposed	5	14	3	54	5(5)	14(14)	3(3)	54(54)	5(5)	14(14)	3(3)	54(54)
$\frac{47}{128}$	[18]	5	3	6	31	6(6)	5(5)	7(7)	45(45)	6(6)	7(7)	5(5)	51(51)
	Proposed	6	7	9	36	6(6)	7(7)	9(9)	36(36)	6(6)	7(7)	9(9)	36(36)
$\frac{83}{128}$	[18]	4	4	6	31	6(6)	5(5)	7(7)	45(45)	8(8)	5(5)	5(5)	51(51)
	Proposed	7	3	6	27	7(7)	3(3)	6(6)	27(27)	7(7)	3(3)	6(6)	27(27)
$\frac{100}{128}$	[18]	3	3	4	30	5(9)	4(4)	5(5)	49(71)	11(13)	4(4)	3(1)	77(83)
	Proposed	13	4	1	48	13(13)	4(4)	1(1)	48(48)	13(13)	4(4)	1(1)	48(48)
$\frac{113}{128}$	[18]	4	4	6	36	7(11)	4(4)	7(7)	55(77)	13(15)	4(4)	5(3)	83(89)
	Proposed	15	5	4	57	15(15)	5(5)	4(4)	57(57)	15(15)	5(5)	4(4)	57(57)

E_r: Error rates, $n_s(n_b)$: the number of consumed sample (buffer) droplets, n_w: the number of waste droplets, e_t: (mixing time + sensing time) in sec, *: ideal scenario.

TABLE VII
VARIATION OF ERROR-TOLERANCE ($\frac{x}{2^{n+1}}$) LIMIT FOR SYNTHETIC TEST-CASES.

Target-*CF*	x=1				x=2				x =3				x=4			
	n_s	n_b	n_m	n_d	n_s	n_b	n_m	n_d	n_s	n_b	n_m	n_d	n_s	n_b	n_m	n_d
$\frac{47}{128}$	6	7	13	4	5	5	9	4	5	4	8	4	5	4	8	4
$\frac{65}{128}$	9	10	18	16	5	7	11	8	3	7	9	4	3	6	8	4
$\frac{99}{128}$	14	5	18	16	8	4	11	8	7	5	11	2	5	4	8	2
$\frac{105}{128}$	14	6	19	16	8	4	11	8	6	4	9	2	5	4	8	2

$n_s(n_b)$: the number of sample (buffer) droplets, n_m: the number of (1:1) mix-split steps, n_d: the number of generated target-droplets.

So far, we have assumed a very stringent constraint on error-tolerance: the error in the desired target-*CF* must be less than $\frac{0.5}{2^n} \times x$, where $x = 1$. However, in real-life biochemical experiments, error-tolerance may be relaxed, i.e, x may be chosen as greater than one [42]. In our experiments, we vary x from 1 to 4 and compute reactant-cost, generated target-droplets, and (1:1) mix-split steps. As evident from Table VII, the cost decreases noticeably when x is increased. Moreover, it also reduces the number of (1:1) mix-split steps, which in turn, decreases sample-preparation time.

VII. CONCLUSIONS

In this work, we have proposed a reliable method for sample preparation on a digital microfluidic lab-on-chip. Instead of explicitly checking whether an error has occurred during split-operations (e.g., by means of sensors) and re-computing all steps since the last checkpoint in these cases, we employ a new *first-try-right* scheme, which is oblivious to such errors. More precisely, we exploit the fact that certain combinations of errors cancel out the unwanted effects. By arranging the mix-split operations in a pre-determined fashion, this yields a procedure where always the correctness of *CF* (within the assumed tolerance) is guaranteed regardless of the presence of any split-error. In order to realize this idea, we have rigorously analyzed the effect of errors on the *CF* of a sample to be prepared. Based on that, new algorithms have been proposed, which, given a *CF*, automatically generate an error-oblivious, yet reliable, mixing graph. An experimental evaluation confirmed the benefits of the proposed methodology. Besides that, the proposed approach can handle multiple split-errors as well as different classes of volumetric errors. As evident from experimental results, it additionally allows for a reduction in the number of consumed sample and buffer droplets as well as mix-split operations compared to previously proposed approaches, when split-errors occur. That is, the proposed method can be performed on a *DMFB* without cyber-physical extensions such as sensors, and also with significantly less resources. As a future research objective, one may study the problem of error-oblivious mixture preparation.

REFERENCES

[1] R. B. Fair, “Digital microfluidics: is a true lab-on-a-chip possible?” *Microfluidics and Nanofluidics*, vol. 3, no. 3, pp. 245–281, 2007.

[2] Z. Li, T.-Y. Ho, and K. Chakrabarty, “Optimization of 3D digital microfluidic biochips for the multiplexed polymerase chain reaction,” *ACM Trans. Des. Autom. Electron. Syst.*, vol. 21, no. 2, pp. 25:1–25:27, 2016.

[3] S. K. Cho, H. Moon, and C.-J. Kim, “Creating, transporting, cutting, and merging liquid droplets by electrowetting-based actuation for digital microfluidic circuits.” *Journal of Microelectromechanical Systems.*, vol. 12, no. 1, pp. 70–80, 2003.

[4] F. S. Fei and K. Chakrabarty, “High-level synthesis of digital microfluidic biochips,” *ACM Journal on Emerging Technologies in Computing Systems*, vol. 3, no. 4, p. 1, 2008.

[5] Z. Li *et al.*, “Droplet size-aware high-level synthesis for micro-electrode-dot-array digital microfluidic biochips,” *IEEE Trans. BioCAS*, vol. 11, no. 3, pp. 612–626, 2017.

[6] O. Keszocze, R. Wille, T.Y.-Ho, and R. Drechsler, “Exact one-pass synthesis of digital microfluidic biochips,” in *Proc. of DAC*. ACM, 2014, pp. 1–6.

[7] R. Wille, O. Keszocze, R. Drechsler, T. Boehnisch, and A. Kroker, “Scalable one-pass synthesis for digital microfluidic biochips,” *IEEE Design & Test*, vol. 32, no. 6, pp. 41–50, 2015.

[8] W. Thies, J. Urbanski, T. Thorsen, and S. Amarasinghe, “Abstraction layers for scalable microfluidic biocomputing.” *Natural Computing*, vol. 7, no. 2, pp. 255–275, 2008.

[9] S. Roy, B. B. Bhattacharya, and K. Chakrabarty, “Optimization of dilution and mixing of biochemical samples using digital microfluidic biochips.” *IEEE Trans. CAD*, vol. 29, no. 11, pp. 1696–1708, 2010.

[10] ——, “Waste-aware dilution and mixing of biochemical samples with digital microfluidic biochips,” in *Proc. of DATE*, 2011, pp. 1059–1064.

[11] J. D. Huang, C. H. Liu, and H. S. Lin, “Reactant and waste minimization in multitarget sample preparation on digital microfluidic biochips.” *IEEE Trans. CAD*, vol. 32, no. 10, pp. 1484–1494, 2013.

[12] S. Bhattacharjee, R. Wille, J.-D. Huang, and B. B. Bhattacharya, “Storage-aware sample preparation using flow-based microfluidic labs-on-chip,” in *Proc. of DATE*, 2018, pp. 1399–1404.

[13] D. Mitra, S. Roy, S. Bhattacharjee, K. Chakrabarty, and B. B. Bhattacharya, “On-chip sample preparation for multiple targets using digital microfluidics,” *IEEE Trans. CAD*, vol. 33, no. 8, pp. 1131–1144, 2014.

[14] S. Roy, B. B. Bhattacharya, S. Ghoshal, and K. Chakrabarty, “Theory and analysis of generalized mixing and dilution of biochemical fluids using digital microfluidic biochips,” *ACM JETC*, vol. 11, no. 1, pp. 2:1–2:33, 2014.

[15] Z. Li *et al.*, “Structural and functional test methods for micro-electrode-dot-array digital microfluidic biochips,” *IEEE Trans. CAD*, vol. 37, no. 5, pp. 968–981, 2018.

[16] T. Xu and K. Chakrabarty, “Fault modeling and functional test methods for digital microfluidic biochips,” *IEEE Trans. BioCAS*, vol. 3, no. 4, pp. 241–253, 2009.

[17] A. Drygiannakis, A. G. Papathanasiou, and A. G. Boudouvis, “On the connection between dielectric breakdown strength, trapping of charge, and contact angle saturation in electrowetting,” *Langmuir*, vol. 25, no. 1, pp. 147–152, 2009.

[18] S. Poddar, S. Ghoshal, K. Chakrabarty, and B. B. Bhattacharya, “Error-correcting sample preparation with cyberphysical digital microfluidic lab-on-chip,” *ACM Trans. Des. Autom. Electron. Syst.*, vol. 22, no. 1, pp. 2:1–2:29, 2016.

[19] M. Ibrahim and K. Chakrabarty, "Error recovery in digital microfluidics for personalized medicine," in *Proc. of DATE*, 2015, pp. 247–252.

[20] A. W. Solomon *et al.*, "A diagnostics platform for the integrated mapping, monitoring, and surveillance of neglected tropical diseases: Rationale and target product profiles." *PLoS Neglected Tropical Diseases*, vol. 6, no. 7, 2002.

[21] W. G. Lee, Y.-G. Kim, B. G. Chung, U. Demirci, and A. Khademhosseini, "Nano/Microfluidics for diagnosis of infectious diseases in developing countries," *Advanced Drug Delivery Reviews*, vol. 62, no. 4-5, pp. 449–457, 2010.

[22] Y. Zhao, T. Xu, and K. Chakrabarty, "Integrated control-path design and error recovery in the synthesis of digital microfluidic lab-on-chip." *ACM JETC*, vol. 6, no. 3, pp. 11:1–11:28, 2010.

[23] Y. Luo, K. Chakrabarty, and T. Y. Ho, "Error recovery in cyberphysical digital microfluidic biochips." *IEEE Trans. CAD*, vol. 32, no. 1, pp. 59–72, 2013.

[24] ——, "Real-time error recovery in cyberphysical digital-microfluidic biochips using a compact dictionary," *IEEE Trans. CAD*, vol. 32, no. 12, pp. 1839–1852, 2013.

[25] Y. L. Hsieh, T. Y. Ho, and K. Chakrabarty, "Biochip synthesis and dynamic error recovery for sample preparation using digital microfluidics," *IEEE Trans. CAD*, vol. 33, no. 2, pp. 183–196, 2014.

[26] M. Alistar, E. Maftei, P. Pop, and J. Madsen, "Synthesis of biochemical applications on digital microfluidic biochips with operation variability," in *Proc. of DTIP (MEMS/MOEMS)*, 2010, pp. 350–357.

[27] M. Alistar, P. Pop, and J. Madsen, "Online synthesis for error recovery in digital microfluidic biochips with operation variability," in *Proc. of DTIP (MEMS/MOEMS)*, 2012, pp. 53–58.

[28] J. Yoshida, *Flash Chemistry: Fast Organic Synthesis in Microsystems*. John Wiley & Sons, Ltd, 2008.

[29] S. Bhattacharjee, S. Poddar, S. Roy, J. D. Huang, and B. B. Bhattacharya, "Dilution and mixing algorithms for flow-based microfluidic biochips," *IEEE Trans. CAD*, vol. 36, no. 4, pp. 614–627, 2017.

[30] J. Gong and C.-J. Kim, "All-electronic droplet generation on-chip with real-time feedback control for EWOD digital microfluidics," *Lab on a Chip*, vol. 8, no. 6, pp. 898–906, 2008.

[31] D. Huijiang *et al.*, "Accurate dispensing of volatile reagents on demand for chemical reactions in EWOD chips," *Lab on a Chip*, vol. 12, no. 18, pp. 3331–3340, 2012.

[32] S. W. Walker and B. Shapiro, "Modeling the fluid dynamics of electrowetting on dielectric (ewod)," *Journal of Microelectromechanical Systems*, vol. 15, no. 4, pp. 986–1000, 2006.

[33] C. Jin, X. Xiong, P. Patra, R. Zhu, and J. Hu, "Design and simulation of high-throughput microfluidic droplet dispenser for lab-on-a-chip applications," in *Proc. of COMSOL*, 2014, pp. 1–7.

[34] Y. Zhao and K. Chakrabarty, "Cross-contamination avoidance for droplet routing in digital microfluidic biochips," *IEEE Trans. CAD*, vol. 31, no. 6, pp. 817–830, 2012.

[35] H. Ren, V. Srinivasan, and R. B. Fair, "Design and testing of an interpolating mixing architecture for electrowetting-based droplet-on-chip chemical dilution," in *Proc. of Solid-State Sensors, Actuators and Microsystems (TRANSDUCERS)*, vol. 1, 2003, pp. 619–622.

[36] Z. Li *et al.*, "Efficient and adaptive error recovery in a micro-electrode-dot-array digital microfluidic biochip," *IEEE Trans. CAD*, vol. 37, no. 3, pp. 601–614, 2018.

[37] Z. Li, K. Y. T. Lai, K. Chakrabarty, T. Y. Ho, and C. Y. Lee, "Droplet size-aware and error-correcting sample preparation using micro-electrode-dot-array digital microfluidic biochips," *IEEE Trans. BioCAS*, vol. 11, no. 6, pp. 1380–1391, 2017.

[38] S. Roy, B. B. Bhattacharya, S. Ghoshal, and K. Chakrabarty, "High-throughput dilution engine for sample preparation on digital microfluidic biochips," *IET Computers Digital Techniques*, vol. 8, no. 4, pp. 163–171, 2014.

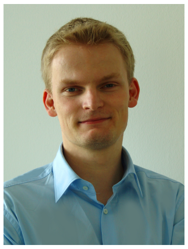
[39] "Supplementary material," 2018. [Online]. Available: https://drive.google.com/open?id=1kvkpFGMvvNX4x8TrmhWOUeCNCE_cLGwk

[40] Bio-protocol, 2017, accessed: June, 2017. [Online]. Available: http://www.bio-protocol.org/Default.aspx

[41] C. Dong, Y. Jia, J. Gao, T. Chen, P.-I. Mak, M.-I. Vai, and R. P. Martins, "A 3D microblade structure for precise and parallel droplet splitting on digital microfluidic chips," *Lab Chip*, vol. 17, pp. 896–904, 2017.

[42] Y. C. Lei, Y. L. Chen, and J. D. Huang, "Reactant cost minimization through target concentration selection on microfluidic biochips," in *Proc. of BioCAS*, 2016, pp. 58–61.

Sudip Poddar received the B.Tech. degree in computer science and engineering from the Maulana Abul Kalam Azad University of Technology (formerly known as West Bengal University of Technology), West Bengal, India, in 2008. He received the M.Tech degree in computer science and engineering from the University of Kalyani, India, in 2012. He is currently working as a Council of Scientific & Industrial Research (CSIR) Research Associate at the Indian Statistical Institute, Kolkata, and a Doctoral candidate at the Indian Institute of Engineering Science and Technology (IIEST), Shibpur, India. His research interests include computer-aided design for microfluidic lab-on-chip and soft computing.



Rober Wille (M'06–SM'09) is Full Professor at the Johannes Kepler University Linz, Austria. He received the Diploma and Dr.-Ing. degrees in computer science from the University of Bremen, Germany, in 2006 and 2009, respectively. He was with the University of Bremen from 2006 to 2015 and has been with the German Research Center for Artificial Intelligence (DFKI), since 2013. He was a Lecturer with the University of Applied Science Bremen and Visiting Professor with the universities of Potsdam and Dresden, Germany. Since 2015, he is working in Linz. His current research interests include design of circuits and systems for both conventional and emerging technologies. He has published over 200 papers in journals and conferences in the above areas. Dr. Wille has served in Editorial Board and Program Committee of numerous journals/conferences, such as IEEE TCAD, DAC, ICCAD, DATE and ASP-DAC.

Hafizur Rahaman (SM 2010) is full professor at the Indian Institute of Engineering Science and Technology (IIEST), Shibpur, India. He was a recipient of Indian National Science Academy (INSA) Visiting Fellowship (2006-2007), Post-Doctoral Fellowship of the Engineering and Physical Sciences Research Council (EPSRC), UK (2006-2007), and Royal Society International Fellowship Award (2008-2009) that provided him support to conduct research for two years at the University of Bristol, United Kingdom. During 2013-2015, he received Senior Research Fellowship to visit the University of Bremen as Visiting Professor. His research interests include design and testing of integrated circuits, biochips, emerging technologies and quantum computing. He has published more than 300 research articles in archival journals and refereed conference proceedings.

Bhargab B. Bhattacharya (F'07) is currently Honorary Visiting Professor of Computer Science and Engineering at the Indian Statistical Institute (ISI), Kolkata, India, and AICTE-INAE Distinguished Visiting Professor. He received the Ph.D. degree in computer science from the University of Calcutta in 1986, and had been on the standing faculty of ISI Kolkata during 1982-2018. He held visiting professorship at the University of Nebraska-Lincoln, and at Duke University, USA, at the University of Potsdam, Germany, at the Kyushu Institute of Technology, Iizuka, Japan, at Tsinghua University, Beijing, China, at IIT Kharagpur and IIT Guwahati, India. His research interest includes design and testing of integrated circuits, nano-biochips, digital geometry, and image processing architecture. He has published more than 380 technical articles, and he holds 10 United States Patents. He is a Fellow of the Indian National Academy of Engineering (INAE) and a Fellow of the National Academy of Sciences, India (NASI). He received the INAE Outstanding Teachers Award (2014) and was named INAE Chair Professor (2016-2018).