Reactant Minimization during Sample Preparation on Digital Microfluidic Biochips using Skewed Mixing Trees

Juinn-Dar Huang
Dept. of Electronics Engineering
National Chiao Tung University
1001 University Road,
Hsinchu 300, Taiwan
jdhuang@mail.nctu.edu.tw

Chia-Hung Liu
Dept. of Electronics Engineering
National Chiao Tung University
1001 University Road,
Hsinchu 300, Taiwan
HSNUskywind@gmail.com

Ting-Wei Chiang
EECS Undergraduate Honors Program
National Chiao Tung University
1001 University Road,
Hsinchu 300, Taiwan
qvc00321@gmail.com

ABSTRACT

Sample preparation is an indispensable process to biochemical reactions. Original reactants are usually diluted to the solutions with desirable concentrations. Since the reactants, like infant's blood, DNA evidence collected from a crime scene, or costly reagents, are extremely valuable, the usage of reactant must be minimized in the sample preparation process. In this paper, we propose the first reactant minimization approach, *REMIA*, during sample preparation on digital microfluidic biochips (DMFBs). Given a target concentration, *REMIA* constructs a skewed mixing tree to guide the sample preparation process for reactant minimization. Experimental results demonstrate that *REMIA* can save about 31%~52% of reactant usage on average compared with three existing sample preparation methods. Besides, *REMIA* can be extended to tackle the sample preparation problem with multiple target concentrations, and the extended version also successfully decreases the reactant usage further.

Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aids

General Terms

Algorithms, Performance, Design

Keywords

Biochip, digital microfluidic biochip (DMFB), dilution, sample preparation, reactant minimization, mixing tree.

1. INTRODUCTION

Lab-on-a-chip (LoC), a kind of biochip, has been becoming one of the emerging research topics in recent years [1][2]. Through integrating various functions in a small chip, LoCs can replace bulky and expensive on-site biochemical systems in labs, hospitals, research centers, or even at home. As the technology has been advancing these years, newly-developed biochips drive fluid droplets across the chip via the electrowetting-on-dielectrics (EWOD) effect [3]. An EWOD-based biochip relies on this effect to dispense, merge, split, transport, and mix droplets through voltage control on the electrodes below the chip [3]–[6]. Since dispensed reactants are all discrete droplets in the chip, this kind of chip is also called digital microfluidic biochip (DMFB). Since a versatile DMFB is requested to carry out a series of associated chemical and biological reactions, a set of automation algorithms, such as synthesis and routing [7]–[12], are essential for speeding up the design process, reducing the manual effort, and improving the design quality. Therefore, the design automation for DMFB has been becoming an emerging topic recently, and the related research works are proliferating.

Sample preparation plays an essential role in biological and chemical reactions, which require reactants to be diluted to the ones with definite concentration values [13]–[15]. For instance, the Trinder's reaction, a common reaction in clinical biochemistry, needs a dilution factor of 200+ to perform an enzymatic glucose assay [14]. Raw reactants are mixed with buffer solutions in a proper order to achieve the target concentration. There are two major concerns during sample preparation: the number of dilution operations and the overall reactant usage. The count of dilution operations basically determines the preparation time and thus should be minimized. A long preparation process may be fatal in some urgent clinical incidents. The other need for sample preparation is to minimize the overall reactant usage. It is particularly true when a reactant is extremely valuable; for instance, a blood sample from a premature baby, a DNA sample collected from a crime scene, or simply a costly reagent.

A reactant may be a physiological sample and its amount is restricted or limited, like infant's blood and DNA evidence. If an excessive amount of blood is drawn from an infant, he or she risks a hospital-induced anemia [16]. Similarly, for a fingerprinting method (e.g., the RFLP method) that typically consumes a large amount of DNA samples, the analysis may fail if the preparation process does not take the issue of sample minimization seriously enough [17]. Meanwhile, for some other assays, the amount of physiological sample is not an issue but the consumption of expensive reagent is. It is not unusual that a routine biochemical assay is repeated numerous times on a daily basis. Consequently, minimizing the reagent consumption in each assay still provides a considerable cost saving, especially for pricey reagents. All in all, minimizing the usage of reactant, either physiological sample or costly reagent, in a biochemical assay is highly beneficial and thus should be encouraged. Without loss of generality, we assume the physiological sample is the most precious reactant (i.e., the minimization target) in the rest of this paper. In contrast, if the amount of sample is not an issue and the buffer solution is instead considered more valuable, it is absolutely fine to regard the usage of buffer solution as the minimization objective for cost saving.

Several approaches aiming at the sample preparation problem have been proposed in recent years [18]–[24]. The most well-known approaches include the bit-scanning (BS) method [20], the algorithm for dilution and mixing with reduced wastage (DMRW) [21], and the improved dilution/mixing algorithm (IDMA) [22]. The BS method leads to a minimal number of dilution operations,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IEEE/ACM International Conference on Computer-Aided Design (ICCAD) 2012, November 5-8, 2012, San Jose, California, USA

Copyright © 2012 ACM 978-1-4503-1573-9/12/11... $15.00

but consumes a large amount of samples and/or buffers. On the other hand, DMRW and IDMA try to minimize the amount of waste droplets. They both suggest that minimizing wastes tends to minimize the usage of samples and buffers. However, in our opinion, waste minimization does not necessarily guarantee that the usage of the most valuable reactant (either physiological sample or reagent) is best minimized. Moreover, it cannot ensure the count of dilution operations is properly reduced, either.

Hence, in this paper, we propose a dedicated reactant minimization algorithm, *REMIA*, for sample preparation on DMFBs. To the best of our knowledge, *REMIA* is the first algorithm regarding the reactant usage as the primary optimization objective. *REMIA* utilizes a skewed mixing tree for reactant minimization during dilution operations. Compared with the three existing methods mentioned previously, *REMIA* saves 31% to 52% of sample usage on average. Moreover, *REMIA* also reduces 19% to 28% of dilution operations on average compared with DMRW and IDMA. Furthermore, *REMIA* can also be applied even if multiple target concentration values are requested simultaneously. Experimental results show that *REMIA* outperforms all the aforementioned methods in terms of sample usage, dilution operation count, and the number of waste droplets.

The rest of this paper is organized as follows. Section 2 describes the dilution process and the mixing models. Section 3 briefly introduces several existing sample preparation techniques. Section 4 shows our motivation and then problem formulation. Section 5 presents our reactant minimization algorithm, *REMIA*, in detail. Experimental results are then reported and discussed in Section 6. Finally, Section 7 concludes this paper.

2. SAMPLE PREPARATION PROCESS

2.1 Serial dilution

During sample preparation, a specified target concentration value (C_t) is achieved via a series of dilution operations. Linear dilution and serial dilution are two dilution procedures commonly used in biochips [13][14]. However, only the serial dilution can be easily carried out on DMFBs since fluids are dispensed as discrete droplets instead of continuous liquid flows. The serial dilution is a logarithmic method that dilutes a stock solution repeatedly using a fixed mixing ratio, like 1:1. For example, if C_t = 25%, the serial method first dilutes a raw sample droplet with a buffer droplet to get a solution with 50% concentration. Then the resultant mixture (intermediate solution or intermediate droplet on DMFB) is again diluted with a buffer droplet to achieve the target concentration. The whole dilution procedure needs two individual dilution operations in this case. Note that a considerable operation count is not uncommon in a serial dilution process [14]. Since the number of dilution operations basically determines the required time of sample preparation, minimizing the dilution operation count thus becomes essential in the sample preparation process.

2.2 Mixing models

Various mixing ratios (or mixing models) are utilized on various DMFB architectures during dilution. There are three different mixing models according to the ratio between two given substances. Assume the ratio between two substances is $(x{:}y)$, where x and y are positive integers, the three models can be expressed as: 1) $x = y = 1$; 2) $x = y \neq 1$; 3) $x \neq y$. DMRW and IDMA adopt the second mixing model through a rotary mixer to reduce the mixing time. However, the rotary mixer is a specially designed circuit block and occupies significant chip area. For a general DMFB design, only the first mixing model can be easily carried out via a linear or an array mixer [4]. To make our work applicable to most common DMFBs without the use of specially-designed rotary mixers, we decide to adopt the (1:1) mixing model and thus estimate the dilution time according to the number of (1:1) mixing operations, just as the BS method does [20].

Nevertheless, an inherent error between the ideal target concentration and the implementable one could exist if the (1:1) mixing model is used. This error is inevitable if the denominator of the target concentration is not a power of 2 or even not a rational number, such as 3 or $\sqrt{2}$. Precisely, the error is limited within $\frac{1}{2^{n+1}}$ if n fractional bits are used to represent the target concentration. Designers can decide how many bits should be taken to ensure an acceptable resultant quantization error [20][21].

2.3 Exponential and interpolated dilution

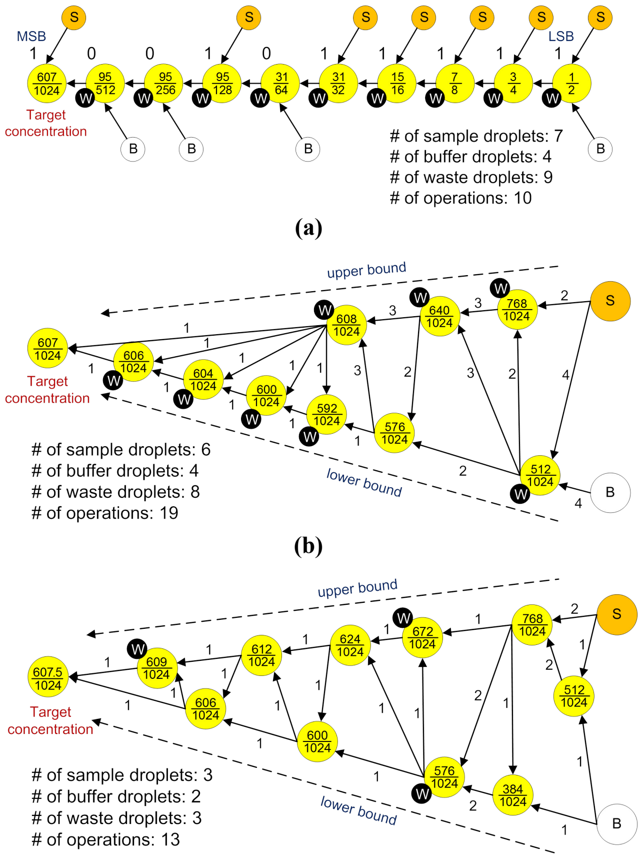
Each dilution operation produces two identical droplets through a mixing operation followed by a splitting operation [14]. The concentration of the resultant droplet after dilution under the (1:1) mixing model can be calculated as $C_r = \frac{C_1+C_2}{2}$, where C_1 and C_2 represent the concentration values of two source droplets before mixing. If a droplet with concentration C is diluted with buffer, the concentration of the resultant droplet becomes $\frac{C}{2}$. Keep diluting the resultant droplet with buffer and repeat the same process by n–1 more times, the concentration of the final resultant droplet eventually becomes $\frac{C}{2^n}$. This kind of dilution, in which buffer is always involved, is called exponential dilution because the concentration of the original droplet decays exponentially. In this paper, a concentration value that can be obtained from a series of exponential dilutions starting from a raw reactant droplet is called a prime concentration value (pcv), e.g., $\frac{1}{32}$. A pcv contains exactly one bit ‘1’ in its binary representation. In contrast, a dilution in which neither of its two source droplets is a buffer droplet is called an interpolated dilution. Exponential dilution and interpolated dilution are both essential to achieve an arbitrary target concentration.

3. PREVIOUS WORKS

To the best of our knowledge, the first algorithm designed for dilution control on biochip was proposed in [18]. It is based on a binary search to decide the concentrations of intermediate droplets and achieves the target concentration through serial dilution. Recent works addressing the sample preparation problem include BS [20], DMRW [21], IDMA [22], intermediate droplet sharing algorithm (IDSA) [23], and ratioed mixing algorithm (RMA) [24]. Among them, BS, DMRW, and IDMA, address the problem of single-target sample preparation. Hence, we briefly describe these three methods in this section.

3.1 Bit-scanning (BS) approach

The BS approach regards a target concentration as a binary string to guide the entire dilution sequence. A bit ‘1’ in the string indicates that a sample droplet should be dispensed for mixing, and a bit ‘0’ requests for a buffer droplet instead. For example, if $C_t = \frac{607}{1024} = 0.1001011111_2$, the dilution sequence follows the binary sequence from the least significant bit (LSB) to the most significant bit (MSB) as shown in Figure 1(a). The length of string determines the number of dilution operations. An inherent nature of the BS approach is that only one intermediate droplet needs to be kept at all times. Once a dilution operation is completed, one of the two resultant droplets is kept, and the other is discarded. The remaining droplet is then turned into one of the two source



(a)

(b)

(c)

Figure 1. Dilution processes for $C_t = \frac{607}{1024} = 0.1001011111_2$ using (a) BS (b) DMRW (c) IDMA

droplets for the next dilution operation. According to this nature, BS doesn't need an extra space for storing a large amount of intermediate droplets, and the chip area can thus be reduced. However, it also implies there is no chance for droplet sharing in BS. Hence, the BS method consumes more samples and buffers as well as produces more wastes as compared to other approaches.

3.2 DMRW and IDMA approaches

DMRW is another method for sample preparation and is the first one considering intermediate droplet sharing for waste droplet minimization. DMRW approaches the target concentration based on a binary search strategy. DMRW keeps track of the lower and upper bounds during the entire search process and the two bounds are initially set as 0 and 1 (i.e., buffer and sample). In each iteration, the lower/upper bound is updated with the average of the two if the target is larger/smaller than the average. The process is not terminated until the target is achieved. Since droplet sharing is considered in DMRW, an extra chip space is required for storing intermediate droplets. The dilution sequence through DMRW is illustrated in Figure 1(b), where the number beside an edge indicates the amount of required droplets, and every black node represents a waste droplet. In this case, 6 sample droplets and 4 buffer droplets are required in the entire dilution process. Compared with BS, DMRW generally consumes few sample and buffer droplets, but spends more time for sample preparation.

IDMA is an improved version of DMRW. For some target concentrations, like $\frac{127}{1024}$ and $\frac{513}{1024}$, the mixing graph produced by DMRW is extremely unbalanced. In such cases, more reactants (sample and buffer) are required, and more wastes are produced. IDMA would moderately relax the bound under certain conditions to avoid producing extremely unbalanced graphs. For example, IDMA turns the mixing graph in Figure 1(b) into the one in Figure 1(c). The lower bound is relaxed from $\frac{512}{1024}$ to 0, which results in a more balanced graph and thus less waste droplets. However, IDMA makes improvements for just a few cases. For most cases, IDMA even consumes more reactants, produces more wastes, and requires more mixing operations. Moreover, IDMA cannot guarantee to always achieve the target concentration, just as the case in Figure 1(c) shows ($\frac{607.5}{1024}$ instead of $\frac{607}{1024}$). It implies that IDMA may induce a larger error than other existing approaches.

The aforementioned algorithms, including DMRW, IDMA, and ISDA, are all developed for waste minimization. The authors claim that: 1) minimizing waste is likely to reduce the usage of samples or expensive reagents; 2) minimizing waste tends to decrease the number of mixing operations due to intermediate droplet sharing [21]–[23]. However, in our opinion, valuable reactants, like blood of a premature baby or expensive reagent, are generally much more valuable than buffer and thus should be prioritized. It is obvious that the methods in [21]–[23] cannot minimize the usage of valuable reactant well because they treat all reactants (sample and buffer here) in an equal way. Consequently, it strongly suggests that a dedicated algorithm concentrating on valuable reactant minimization is essential for DMFB-based applications indeed. Moreover, we also agree that the number of (1:1) mixing operations can better estimate the overall sample preparation time in a general DMFB without specially designed rotary mixers. The experimental results further demonstrate that our reactant minimization algorithm (named *REMIA*) achieves a lower count of mixing operations as compared to [21][22].

4. MOTIVATION

4.1 Mixing tree

In this paper, we model a dilution process using a tree structure, named mixing tree. The mixing tree shown in Figure 2 illustrates the same dilution sequence depicted in Figure 1(a). Each node v in a mixing tree is associated with a concentration value $\text{cv}(v)$, $0 \leq \text{cv}(v) \leq 1$. The edges indicate the dependency of dilution operations. cv(r) is set to the target concentration C_t for the root node r. In the BS method, cv(l) is either 1 or 0 for every leaf node l because one of two source droplets for dilution is always a sample or buffer droplet.

In a dilution operation, two source droplets are always required, and thus the out-degree of a branch (non-leaf) node in a mixing tree is always two. That is, a mixing tree must be a *full* binary tree, in which every branch node has exactly two children. Besides, since only one of two resultant droplets is used for the next dilution, every branch node except the root node in a mixing tree implies a waste droplet. As well, under the (1:1) mixing

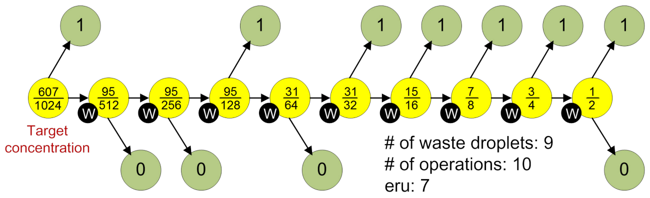


Figure 2. The mixing tree of the dilution process in Figure 1(a)

model, the concentrations of a branch node v_p and its two children v_l and v_r are related, and the relationship can be expressed as:

$$\mathrm{cv}(v_p) = \frac{\mathrm{cv}(v_l) + \mathrm{cv}(v_r)}{2} \tag{1}$$

Besides, the number of dilution operations and waste droplets in a mixing tree T can be given as follows:

$$\#\mathrm{branch\_node}(T) = \#\mathrm{operation}(T) = \#\mathrm{waste}(T) + 1 \tag{2}$$

Meanwhile, the total usage of valuable reactant cannot be easily obtained for a mixing tree T since the way of how its leaf nodes are produced remains unknown. However, the minimal requirement of reactant can still be found. Since the concentration of a droplet specifies the amount of reactant it contains, the sum of the concentrations of all leaf nodes gives a lower bound for the reactant requirement. If the sum is not an integer, it must be rounded up. The final value after rounding is called essential reactant usage (eru), which can be expressed as:

$$\mathrm{eru}(T) = \left\lceil \sum_{v\ is\ a\ leaf\ node} \mathrm{cv}(v) \right\rceil \tag{3}$$

An optimal exponential dilution process (described later) can guarantee to produce all leaf nodes of a mixing tree T only with the amount of sample equal to $\mathrm{eru}(T)$. For instance, Figure 5 illustrates an optimal process for producing all leaf nodes of the mixing tree T shown in Figure 3, where $\mathrm{eru}(T) = \left\lceil \frac{1856}{1024} \right\rceil = 2$. An optimal algorithm will be presented in Section 5.3 for producing a reactant-minimal exponential dilution sequence.

4.2 Motivation

Both of the mixing trees illustrated in Figure 2 and Figure 3 can produce a resultant droplet with the same target concentration $\frac{607}{1024}$. However, the mixing tree in Figure 2 consumes seven sample droplets, while the mixing tree in Figure 3 merely needs two. The improvement is quite tremendous ($\frac{7-2}{7} \cong 71\%$). This clearly indicates that the difference in the usage of valuable reactant can be very significant between two mixing trees even the target concentration is identical. It inspires us that – given a target concentration, finding a mixing tree with less eru implies the consumption of reactant is better minimized. Therefore, the key part of our reactant minimization algorithm *REMIA* is to construct a mixing tree with minimized eru for a given target concentration.

4.3 Problem formulation

In this paper, the reactant minimization problem is formally formulated as follows. Given a target concentration $0 < C_t < 1$, determine a dilution process under the (1:1) mixing model to produce a resultant droplet with concentration C_t such that the valuable reactant usage is minimized.

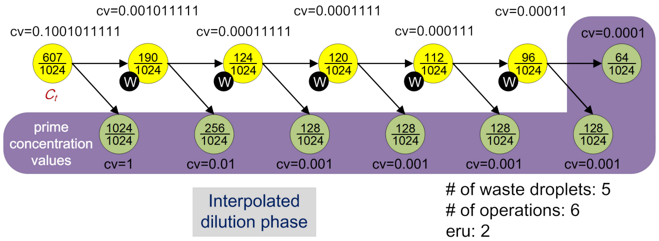


Figure 3. The skewed mixing tree for $C_t = \frac{607}{1024}$

5. PROPOSED ALGORITHM

5.1 Algorithm overview

The proposed reactant minimization algorithm *REMIA* determines the whole dilution process in two phases, the former interpolated dilution phase and the latter exponential dilution phase. In the former phase, a skewed mixing tree is constructed for eru minimization. Then, the latter phase builds an exponential dilution tree for producing all the pcvs required by the leaf nodes of the mixing tree created in the former phase.

5.2 Interpolated dilution phase

In this phase, *REMIA* builds a skewed mixing tree T with minimized eru for a given target concentration C_t. For every leaf node v in T, cv(v) is always a pcv. It implies no buffer droplets are used in this phase, and thus only interpolated dilution operations are performed. The mixing tree T is built in a top-down (i.e., from root to leaves) manner based on a greedy strategy. The method of how the mixing tree is built is outlined in Figure 4. In each iteration, the concentration value *lcv*/*rcv* for the newly created left/right child node is derived from the concentration value *ccv* of the parent node. *lcv* is set as the double of the most significant non-zero bit (MSNZB) of *ccv*. Since *lcv* is always a pcv, every left child node must be a leaf node. On the other hand, *rcv* is set as $2{\times}ccv - lcv$ based on (1) (i.e., $ccv = \frac{lcv+rcv}{2}$). At the end, the right child node is regarded as the new parent node, and the next iteration starts. This process is not terminated until *rcv* eventually becomes a pcv.

If C_t has n non-zero bits in its binary representation, the corresponding mixing tree must have n leaf nodes and cv(v) must be a pcv for every leaf node v. As well, the tree is always a skewed binary tree. Figure 3 shows a tree built in such a way. The given target concentration is $\frac{607}{1024}$, which possesses seven non-zero bits in its binary representation. As a result, the constructed mixing tree has seven leaf nodes and it is inherently skewed as described previously.

Since half of sample is discarded through a waste droplet after a dilution operation, the key idea of the proposed mixing tree building method is to ensure that source droplets with higher concentration are mixed as late as possible to reduce the sample waste. That is, leaf nodes with higher concentration are kept closer to the root for better sample minimization. The experimental results shown in Section 5 indicate that the proposed method can achieve almost the same minimization outcome as compared to an optimal integer linear programming (ILP)-based approach with exponential time complexity. Nevertheless, the time complexity of the proposed method is merely $O(n)$, where n is the number of precision bits.

```
BUILD-MIXING-TREE($C_t$)
    create a node root ; cv(root) ← $C_t$
    cur ← root ; ccv ← $C_t$
    WHILE(ccv is not a pcv)
        create two child nodes left and right for cur
        lcv ← EXTRACT-MSNZB(ccv) × 2    // lcv is a pcv
        rcv ← 2 × ccv – lcv    // lcv + rcv = ccv × 2
        cv(left) ← lcv ; cv(right) ← rcv
        cur ← right ; ccv ← rcv
    return root
```

Figure 4. The proposed method for building a mixing tree

5.3 Exponential dilution phase

The objective of this phase is producing all the leaf nodes required by the given mixing tree. Based on a fact that two droplets of an identical pcv can be produced simultaneously through a single exponential dilution operation, *REMIA* performs a series of exponential dilution operations to create all required nodes of specified pcvs with minimal sample usage.

Exponential dilution trees (EDTs) are used to better describe the exponential dilution phase. Each node v in an EDT represents a droplet with a concentration value of cv(v). The root of an EDT always has a concentration of 1. A directed edge in an EDT points from a source droplet to a resultant droplet in an exponential dilution. Since an exponential dilution produces two resultant droplets, the out-degree of a branch node is at most two. A branch node may have only one child, which implies only one resultant droplet is required for succeeding operations and the other one is thus discarded as a waste. According to the above facts, an EDT is a binary tree but is not necessary a full one. Also, cv(p) = cv(c) ×2 if c is a child of p. As a result, the concentration value of every node must be a pcv because the value of each leaf node is a pcv by definition. Moreover, each branch node in an EDT represents an exponential dilution operation and each operation needs one buffer droplet. Hence, the number of branch nodes is equal to the count of required dilution operations and the number of buffer droplets. On the other hand, the number of waste droplets is equal to the count of single-child branch nodes. Figure 5 demonstrates a complete exponential dilution phase for producing the leaf nodes required by the mixing tree illustrated in Figure 3.

Unlike a mixing tree, an EDT is built in a bottom-up fashion (i.e., from leaves to root) based on a greedy strategy, which is very similar to the well-known Huffman encoding algorithm [25]. There may be more than one EDT being built in the exponential dilution phase. Precisely, the final outcome is an EDT forest instead. The method for building an EDT forest is outlined in Figure 6. Initially, for a given mixing tree T, all its leaf nodes are inserted into a min-heap. In each iteration, a nodes x is extracted from the min-heap. If cv(x) is equal to 1, then x is identified as a root of an EDT. If not, create a new node z, set cv(z) to cv(x)×2, assign x as z's left child, and then extract a node y from the min-heap again. If $cv(x) = cv(y)$, assign y as z's right child; otherwise, insert y back into the min-heap. Lastly, insert z back into the min-heap. This process is not terminated until the min-heap is eventually empty.

The outcome of this phase is a set of EDTs whose cardinality is equal to eru(T). An example is illustrated in Figure 5, in which all the leaf nodes required by the mixing tree shown in Figure 3 are produced through two EDTs. The total number of dilution operations (and waste droplets) can be calculated as the sum of the values from both phases. For instance, the total number of waste droplets is 7 (i.e., 5+2) for $C_t = \frac{607}{1024}$ if *REMIA* is applied.

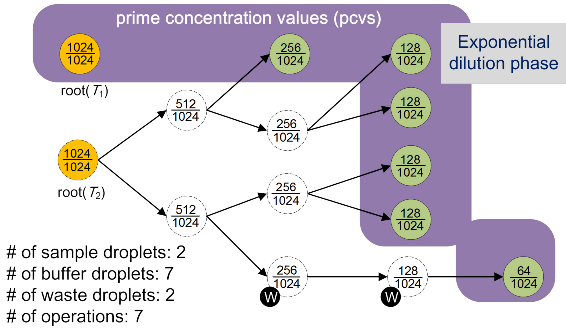


Figure 5. The exponential dilution trees for $C_t = \frac{607}{1024}$

```
BUILD-EDT-FOREST(T)    // T is a mixing tree
    create a min-heap Q containing all leaf nodes of T
    F = ∅    // F is a set of roots of EDTs, initially empty
    WHILE(Q is not empty)
        x ← EXTRACT(Q)
        if(cv(x) ≠ 1) then
            create a node z ; cv(z) ← cv(x) × 2
            left_child(z) ← x
            y ← EXTRACT(Q)
            if(cv(x) = cv(y)) then right_child(z) ← y
            else INSERT(Q, y)
            INSERT(Q, z)
        else F = F ∪ {x}
    return F
```

Figure 6. The proposed method for building an EDT forest

5.4 Extension to multi-target preparation

REMIA can be extended to handle the sample preparation problem of multiple target concentrations (or multi-target preparation for short), which was firstly discussed in [23]. That is, a set of target concentrations should be prepared at the same time. In the extended *REMIA*, the mixing tree of each target concentration is still constructed individually in the interpolated dilution phase. Nevertheless, leaf nodes of all mixing trees are inserted into a single shared min-heap in the exponential dilution phase. In most cases, the unified min-heap can effectively reduce the sample usage because:

$$\left\lceil \sum_T \sum_{v \in leaf(T)} cv(v) \right\rceil \le \sum_T \left\lceil \sum_{v \in leaf(T)} cv(v) \right\rceil \qquad (4)$$

Obviously, the extended *REMIA* is very likely to reduce the total sample usage further. Besides, the total dilution operation count, the total buffer usage, and the total waste count, can all be minimized further. Figure 7 demonstrates a multi-target example, where two target concentrations are $\frac{607}{1024}$ and $\frac{96}{1024}$, respectively. Figure 7 also indicates the specific improvements in all aspects from the unified exponential dilution phase.

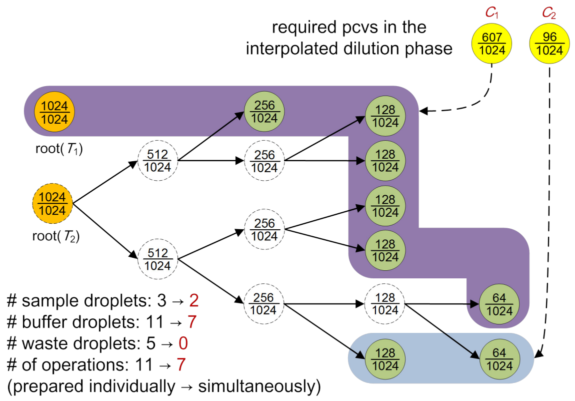


Figure 7. The dilution process for two target concentrations, $C_1 = \frac{607}{1024}$ and $C_2 = \frac{96}{1024}$

6. EXPERIMENTAL RESULTS

To demonstrate the effectiveness of *REMIA*, we compare it against the three aforementioned sample preparation methods – BS, DMRW, and IDMA. Besides, an ILP-based method, which guarantees to produce a mixing tree with minimal sample usage, is also included for comparisons. Two experiments, one for single-target preparation and the other for multi-target preparation, are conducted. We adopt the same experimental environment setup used in [21]–[23], where various target concentrations are in the range between $\frac{1}{1024}$ and $\frac{1023}{1024}$; that is, the number of fractional bits is set to ten.

6.1 Single-target preparation

The experimental results for single-target preparation are given in Table 1. The averages over 1023 different concentration settings are reported. The results indicate that BS has the largest sample usage but the lowest operation count. Waste-aware DMRW and IDMA perform very well in waste amount, moderately in sample usage, but worst in operation count. *REMIA* performs very well in sample minimization, well in operation count, but produces more wastes than DMRW and IDMA. More specifically, *REMIA* consumes 52%/31%/31% less sample as compared with BS/DMRW/IDMA. Though *REMIA* is primarily designed for reactant minimization, it still outperforms DMRW and IDMA by 19% and 28% in operation count. Though other implementation details, such as scheduling, layout, and routing, can also affect the whole processing time, the operation count is still a good indicator for time estimation at early stages. The results clearly demonstrate that minimizing valuable reactant instead of waste can generally achieve a better outcome.

In fact, after examining all 1023 cases, *REMIA* always wins or ties in sample usage for every case as compared with BS, DMRW and IDMA. Besides, *REMIA* always wins or ties in operation count for every case as compared to DMRW and IDMA. Table 2 enumerates some extreme cases, in which *REMIA* makes a significant improvement. The sample usage can be reduced up to 87% (to BS), while the operation count can be reduced up to 57% (to IDMA). Most surprisingly, *REMIA* even outperforms waste-aware DMRW and IDMA in waste amount for certain cases.

Besides, we compare *REMIA* with an ILP-based reactant-minimal method, which guarantees to produce a mixing tree with minimal sample usage. The results show that *REMIA* can do nearly as good as the optimal ILP-based method in terms of reactant minimization (with only 1% difference). In 1010 out of all 1023 cases, *REMIA* reports the identical sample usage as the ILP method. It suggests that *REMIA* is a very good and fast heuristic algorithm for reactant minimization.

6.2 Multi-target preparation

In this experiment, a set of *k* different target concentrations are picked for preparation where *k* = 1, 2, 3, 10, 20, 50, and 100, just as [23]. For each *k*, 1000 randomly chosen combinations are examined to get an average performance. Five approaches, including BS, DMRW, IDMA, original *REMIA*, plus the extended *REMIA*, are evaluated for comparisons. IDSA [23] is not included since it uses a different mixing model. Figure 8 shows the experimental results in terms of sample usage, waste amount, and dilution operation count. Note that the reported data in Figure 8 are on a per-target basis to ease the comparisons. For instance, if a total of 7 dilution operations are required in a two-target preparation process, then the operation count per target is 3.5.

Figure 8 reveals three facts. Firstly, except for the extended *REMIA*, each of the other four methods basically performs equally no matter how many targets there are. It is because that those four methods consider only one target at a time and do not have a simple way to explore any sharing between targets at all. As a consequence, the average results per target are almost unchanged even as the number of targets grows. Secondly, the extended *REMIA* does improve the quality of result (QoR) in all aspects. This should not be a surprise since the extended *REMIA* explores the sharing between different targets in its unified exponential dilution phase. In general, the more the targets are, the better the QoR is. Especially, the average sample usage for a target can even be reduced below two, which is very notable. Thirdly, the extended *REMIA* outperforms all the other four methods in all aspects if the number of targets is two or more. Here, we have no intention to claim that the extended *REMIA* is a complete and dedicated solution for multi-target sample preparation just as

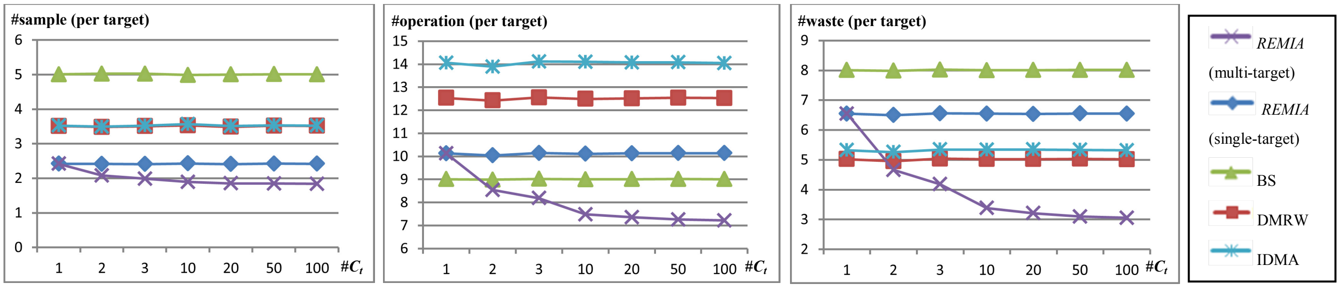
Table 1. Results for single-target preparation

	BS	DMRW	IDMA	ILP	*REMIA*
#sample	5.00	3.51	3.52	2.40	2.42
	(0.48)[*]	(0.69)	(0.69)	(1.01)	(1.00)
#operation	9.01	12.53	14.06	10.30	10.13
	(1.12)	(0.81)	(0.72)	(0.98)	(1.00)
#waste	8.01	5.02	5.32	6.70	6.55
	(0.82)	(1.30)	(1.23)	(0.98)	(1.00)

[*] Numbers in parentheses: the ratios between *REMIA* and the specified methods

Table 2. Extreme cases in single-target preparation

	C_t	#sample	#operation	#waste
(BS, *REMIA*)	191/1024	(7, 1)	(10, 13)	(9, 6)
(DMRW, *REMIA*)	641/1024	(7, 2)	(18, 10)	(9, 8)
(IDMA, *REMIA*)	850/1024	(8, 3)	(21, 9)	(8, 6)



(a) (b) (c)

Figure 8. Average (a) sample (b) operation (c) waste count per target in multi-target preparation

IDSA [23]. Nevertheless, it does provide a good start for further reactant minimization.

7. CONCLUSUIONS AND FUTURE WORK

Sample preparation is an essential process to biochemical reactions. Several previous works have been proposed for waste minimization. However, in our opinion, a reactant (either sample or expensive reagent) can be extremely valuable, and thus its usage should be minimized in the dilution process. Therefore, in this paper, we present the first algorithm for reactant minimization, *REMIA*, for DMFBs. The skewed mixing tree and the exponential dilution tree (EDT) are both introduced to describe the proposed sample preparation process. We also present an extended *REMIA* for multi-target preparation. The extensive experimental results demonstrate that *REMIA* performs better than existing techniques in terms of reactant usage. Furthermore, the extended *REMIA* outperforms the prior arts in terms of reactant usage, waste amount, and dilution operation count. As a result, it is convincing that *REMIA* is indeed a better solution while performing sample reparation on DMFBs.

We have already demonstrated that *REMIA* can be extended to deal with the multi-target preparation problem through a unified exponential dilution phase. On top of the extended *REMIA*, we are currently developing a new algorithm for multi-target preparation, which focuses on reducing the reactant usage in the interpolated dilution phase.

8. REFERENCES

[1] R. Daw and J. Finkelstein, “Insight: Lab on a chip,” *Nature*, vol. 442, no. 7101, pp. 367–418, Jul. 2006.

[2] S. J. Lee and S. Y. Lee, “Micro total analysis system (μ-TAS) in biotechnology,” *Appl. Microbiol. Biotechnol.*, vol. 64, no. 3, pp. 289–299, Jan. 2004.

[3] M. G. Pollack, R. B. Fair, and A. D. Shenderov, “Electrowetting-based actuation of liquid droplets for microfluidic applications,” *Appl. Phys. Lett.*, vol. 77, no. 11, pp. 1725–1726, Sep. 2000.

[4] P. Paik, V. K. Pamula, and R. B. Fair, “Rapid droplet mixers for digital microfluidic systems,” *Lab. Chip*, vol. 3, no. 4, pp. 253–259, Sep. 2003.

[5] V. Srinivasan, V. K. Pamula, and R. B. Fair, “An integrated digital microfluidic lab-on-a-chip for clinical diagnostics on human physiological fluids,” *Lab. Chip*, vol. 4, no. 4, pp. 310–315, May 2004.

[6] T.-Y. Ho, J. Zeng, and K. Chakrabarty, “Digital microfluidic biochips: a vision for functional diversity and more than Moore,” in *Proc. IEEE/ACM ICCAD*, 2010, pp. 578–585.

[7] M. Cho and D. Z. Pan, “A high-performance droplet routing algorithm for digital microfluidic biochips,” *IEEE TCAD*, vol. 27, no. 10, pp. 1714–1724, Oct. 2008.

[8] M. Alistar, E. Maftei, P. Pop, and J. Madsen, “Synthesis of biochemical applications on digital microfluidic biochips with operation variability,” in *Proc. DTIP Symp.*, 2010, pp. 350–357.

[9] L. Luo and S. Akella, “Optimal scheduling of biochemical analyses on digital microfluidic systems,” IEEE *T-ASE*, vol. 8, no. 1, pp. 216–227, Jan. 2011.

[10] C. C.-Y. Lin and Y.-W. Chang, “Cross-contamination aware design methodology for pin-constrained digital microfluidic biochips,” *IEEE TCAD*, vol. 30, no. 6, pp. 817–828, Jun. 2011.

[11] Z. Xiao and E. F. Y. Young, “Placement and routing for cross-referencing digital microfluidic biochips,” *IEEE TCAD*, vol. 30, no. 7, pp. 1000–1010, Jul. 2011.

[12] T.-W. Huang, S.-Y. Yeh, and T.-Y. Ho, “A network-flow based pin-count aware routing algorithm for broadcast-addressing EWOD chips,” *IEEE TCAD*, vol. 30, no. 12, pp. 1786–1799, Dec. 2011.

[13] G. M. Walker, N. Monteiro-Riviere, J. Rouse, A. T. O’Neill, “A linear dilution microfluidic device for cytotoxicity assays,” *Lab. Chip*, vol. 7, no. 2, pp. 226–232, Oct. 2007.

[14] 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. TRANSDUCERS*, 2003, pp. 619–622.

[15] H. Moon, A. R. Wheeler, R. L. Garrell, J. A. Loo, and C.-J. Kim, “An integrated digital microfluidic chip for multiplexed proteomic sample preparation and analysis by MALDI-MS,” *Lab. Chip*, vol. 6, no. 9, pp. 1213–1219, Jul. 2006.

[16] K. A. Kalanick, *Phlebotomy Technician Specialist*, *2nd Edition*. Delmar Cengage Learning, 2011.

[17] J. M. Butler, *Fundamentals of Forensic DNA Typing*. Academic Press, 2009.

[18] E. J. Griffith, S. Akella, and M. K. Goldberg, “Performance characterization of a reconfigurable planar-array digital microfluidic system,” *IEEE TCAD*, vol. 25, no. 2, pp. 345–357, Feb. 2006.

[19] T. Xu, V. K. Pamula, and K. Chakrabarty, “Automated, accurate, and inexpensive solution-preparation on a digital microfluidic biochip,” in *Proc. IEEE BioCAS*, 2008, pp. 301–304.

[20] W. Thies, J. P. Urbanski, T. Thorsen, and S. Amarasinghe, “Abstraction layers for scalable microfluidic biocomputing,” *Nat. Comput.*, vol. 7, no. 2, pp. 255–275, May 2008.

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

[22] S. Roy, B. B. Bhattacharya, and K. Chakrabarty, “Waste-aware dilution and mixing of biochemical samples with digital microfluidic biochips,” in *Proc. IEEE/ACM DATE*, 2011, pp. 1059–1064.

[23] Y.-L. Hsieh, T.-Y. Ho, and K. Chakrabarty, “On-chip biochemical sample preparation using digital microfluidics,” in *Proc. IEEE BioCAS*, 2011, pp. 297–300.

[24] S. Roy, B. B. Bhattacharya, P. P. Chakrabarti, and K. Chakrabarty, “Layout-aware solution preparation for biochemical analysis on a digital microfluidic biochip,” in *Proc. IEEE VLSID*, 2011, pp. 171–176.

[25] D. A. Huffman, “A method for the construction of minimum redundancy codes,” in *Proc. IRE*, 1952, pp. 1098–1101.