

# Cyber–Physical Digital-Microfluidic Biochips: Bridging the Gap Between Microfluidics and Microbiology

By MOHAMED IBRAHIM<sup>1</sup>, Student Member IEEE, AND KRISHNENDU CHAKRABARTY, Fellow IEEE

**ABSTRACT** | Digital microfluidics is transforming microbiology research by providing new opportunities for high-throughput sample preparation and point-of-care diagnostics. Over the past decade, several design-automation (synthesis) techniques have been developed for on-chip droplet manipulation. However, these methods oversimplify the dynamics of biomolecular protocols and they have yet to make a significant impact in biochemistry/microbiology research, leading to a large gap between advances in biochip design and the adoption of biochips for running biomolecular protocols. In this paper, we bridge this gap by introducing a new paradigm for biochip design automation. By exploiting advances in the integration of sensing systems into a digital-microfluidic biochip, we present a number of synthesis solutions that use realistic models of biomolecular protocols to address real-world microbiology applications through cyber–physical adaptation. This paper also details a vision for continued research on design-automation and optimization methodologies for the realization of biomolecular protocols using microfluidic biochips.

**KEYWORDS** | Biochip; biomolecular; cyber–physical; gene-expression analysis; lab-on-chip; microfluidics; synthesis

## I. INTRODUCTION

Microfluidics technology has tremendous potential to advance subject areas ranging from drug discovery to clinical diagnostics through the use of miniaturized devices for biomolecular recognition [1], [2]. The microfluidic industry is now poised to become mainstream as the microfluidic

devices market is forecast to grow from \$2.56 billion in 2015 to \$5.95 billion in 2020 [3]. The rapid evolution of lab-on-chip technologies opens up new opportunities for point-of-care (POC) applications, including clinical diagnostics [4] and electrochemistry [5].

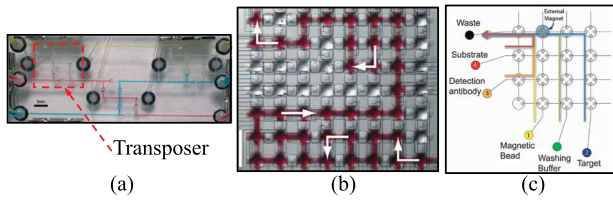
An early generation of microfluidic devices, referred to as continuous-flow microfluidic biochips (CMFs) [6], consists of microfabricated channels, micropumps, and microvalves that are permanently etched in a silicon or a glass substrate. Initially, CMFs relied on simple topologies and only a few channels; however, commercial microfluidic devices today incorporate large-scale networks of channels so that they can be used in implementing complex tasks such as organ-on-a-chip assays [7]. For example, a product from Fluidigm [8], a biotechnology company that produces CMFs, can perform a series of gene analyses steps, including enrichment for a target DNA sequence, sample barcoding, for multiplexed sequencing, and preparation of the sequencing library. While CMFs are highly efficient and robust in the miniaturization and high-throughput execution of biochemical assays, the structure and the functionality of such devices are tightly coupled. As a result, each biochip is only applicable to a narrow class of applications, and the reconfigurability of this early technology is quite limited.

Recent research has focused on making traditional CMF designs more flexible by using valve-based reconfigurable components. For example, Silva *et al.* [9] introduced a programmable flow-based primitive, called a transposer, that can be scaled to create a field-programmable flow-based fabric. Also, a software-programmable microchannel structure has been introduced in [10], which allows a software program to control numerous valves by using microfluidic multiplexer circuitry. Similarly, in [11], Kim *et al.* presented a design of field-programmable

Manuscript received June 10, 2017; revised September 15, 2017; accepted October 1, 2017. This work was supported by the U.S. National Science Foundation under Grants CCF-1702596 and CCF-1135853. (Corresponding author: Mohamed Ibrahim.)

The authors are with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708 USA (e-mail: mohamed.s.ibrahim; krishnendu.chakrabarty@duke.edu).

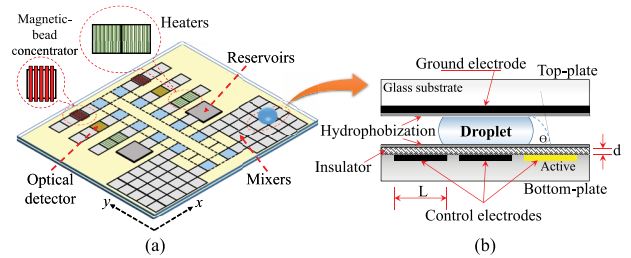
Digital Object Identifier: 10.1109/JPROC.2017.2759251



**Fig. 1. Examples of field-programmable CMFs. (a) A transposer-based reconfigurable CMF [9]. (b) Software-programmable array [10]. (c) Programmable microfluidic platform (PMP) [11].**

flow-based devices, called programmable microfluidic platforms (PMPs). In these devices, fluid can be flexibly manipulated using a 2-D array of valves that can act as both flow-switching elements and reaction chambers. A variety of biochemical applications such as immunoassays and genetic analysis have been implemented using these reconfigurable devices. Fig. 1 shows three examples of field-programmable CMFs. These reconfigurable devices have recently attracted researchers to develop design-automation and testing solutions, thus leveraging the flexibility and programmability provided by these devices [12]–[14].

Despite the advances in field-programmable CMFs, they are still limited in their scalability with respect to valve control and their ability to support both multiple sample pathways and real-time decision making; such characteristics are inherent in today’s biochemical applications. In contrast, a digital microfluidic biochip (DMFB) is a reconfigurable lab-on-a-chip technology that has achieved remarkable success in enabling miniaturized analysis systems for several biochemical applications. Motivated by the precision and cost reduction offered by digital microfluidics (DMF), Illumina, a market leader in DNA sequencing, has recently developed DMFBs for high-quality DNA-library preparation [15]. This technology has also been deployed by GenMark for infectious disease testing [16] and by Baebies for the detection of lysosomal enzymes in newborns [17]. These significant milestones indicate the impressive growth of the digital-microfluidics market and the promising miniaturization capabilities offered by this technology for immunoassays for POC diagnosis [18], [19], DNA sequencing [20], and environmental monitoring [21]. A typical DMFB consists of a 2-D electrode array, on-chip reservoirs, optical sensors, and heaters, as shown in Fig. 2(a). A cell in a DMFB consists of two parallel plates; see Fig. 2(b). The electrode surface is coated with a thin layer of an insulator such as Paralyene [18]. Both plates are also coated with a thin film to provide a hydrophobic platform that is necessary for smooth electrowetting-based droplet actuation [22]. The gap between the top and bottom plates is usually filled with silicon oil, which acts as a filler medium to prevent surface contamination. When an electric field  $V$  is applied between the parallel plates of a DMFB, the interfacial surface energies are modulated and an electrical double layer is created, which in turn alters the apparent contact angle  $\theta(V)$  of a conductive liquid



**Fig. 2. Schematic view of a DMFB. (a) A DMFB with a 2-D array of electrodes. (b) A side view of the DMFB.**

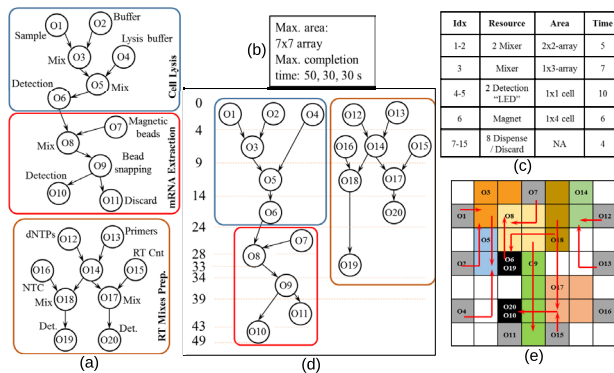
droplet that is in contact with the hydrophobic surface [Fig. 2(b)]. The change in the contact angle, in turn, influences the wetting behavior of the droplet. This phenomenon is known as electrowetting-on-dielectric (EWOD), and it can be modeled using the Lippmann–Young equation [22]

$$\cos(\theta(V)) = \cos(\theta(0)) + \frac{\epsilon_0 \epsilon_r V^2}{2d\gamma_{LG}} \quad (1)$$

where  $\gamma_{LG}$  is the liquid–gas interfacial tension,  $\epsilon_0$  is the permittivity of vacuum,  $\epsilon_r$  is the permittivity of the bottom insulator, and  $d$  is its thickness.

Hence, DMFBs allow biochemical assays to be carried out in a miniaturized form, where picoliter volumes of fluids can be transported, mixed, heated, or analyzed using various means of detection. The electrical actuation of DMFBs enables software control of all these fluid-handling operations in a real-time manner [23]. Benefits of miniaturization using DMFBs include reduced reagent consumption and sample requirement (which is critical in POC setting), reduced analysis time due to the increased reaction speed, human-intervention-free control of droplets via design automation, and low risk of contamination. These benefits make DMFB technology an ideal alternative to conventional benchtop biochemical procedures [24]. Therefore, as the applications of DMFBs grow, greater demands are likely to be placed on developing a complete microfluidic solution for advanced molecular biology research. Furthermore, the flexibility provided by this technology for integrated biomolecular analysis has been supported by advances in integrated sensing technologies [25], magnetic bead manipulation [26], on-chip thermal-cycling procedures [25], and DMFB-enabled electroporation techniques [27]. These advances enable complex bench-top biomolecular procedures on programmable and cyber-physical DMFBs.

Motivated by the increasing complexity of biochemistry-on-chip, research on design automation (or “synthesis”) for DMFBs has received much attention over the past decade [28], [29]. The goals of such design-automation methods are twofold: 1) nonexpert users of DMFBs (e.g., a local clinician in a rural area [30]) should be able to utilize only a high-level specification of a protocol, e.g., a sequencing graph [Fig. 3(a)], to program a biochip; and 2) using information about DMFB-enabled devices [Fig. 3(c)] and prespecified



**Fig. 3. Synthesis flow for a DMFB that processes mRNA samples. (a) Sequencing graph. (b) Design specification. (c) Library of DMFB modules. (d) Scheduled sequencing graph. (e) Droplets control using actuation sequences.**

operation constraints [Fig. 3(b)], a well-established synthesis framework must convert the abstract representation of a protocol into an executable code (namely actuation sequences) that timely actuate the electrodes of a DMFB; thus moving droplets to execute the associated protocol, as shown in Fig. 3(e). While prior synthesis methods address droplet manipulation on a chip for basic fluidic operations, they have yet to address the challenges associated with non-trivial biology-on-chip. A key limitation of prior solutions for biochip synthesis is that they consider a given sequencing graph and therefore a predetermined sequence of fluidic operations. However, in actual biochemistry protocols, the actual sequence of fluidic operations is not known until intermediate reaction results are available [31]. Therefore, biochip designs based on static synthesis are unrealistic and today's techniques do not exploit the potential of DMFBs for implementing real-life microbiology applications. By leveraging advances in cyber-physical integration of DMFBs, an effective synthesis capability can be developed to overcome the myriad complexities of biomolecular protocols and therefore bridge the gap between microfluidics and microbiology [7], [32].

### A. Contributions of the Paper

In this paper, we provide a comprehensive design-automation framework that can support scalable execution of biomolecular protocols using cyber-physical DMFBs. This framework addresses design-automation challenges arising from mapping realistic implementation of quantitative-analysis protocols to the chip scale; such mapping is guided by our recent experience with benchtop experiments from the epigenetics domain [31].

Motivated by our benchtop study, we first pinpoint the main characteristics of quantitative-analysis protocols, whereby these characteristics can be incorporated into the algorithms underlying the design-automation flow.

- 1) Multipathway experimentation with real-time decision making: Quantitative results are reported numerically and are compared against an accompanying reference for interpretation.<sup>1</sup> For this purpose, multiple samples are treated through independent pathways. Furthermore, decision-making and cyber-physical adaptation capability is needed to overcome the inherent uncertainty about the order of basic fluidic steps for each sample.
- 2) Biomolecular analysis under temporal constraints: Temporal constraints may arise during protocol execution due to physical phenomena such as droplet evaporation or deadlines imposed by the target chemistry, e.g., degradation of samples and reagents [33].
- 3) Differentiation and high-throughput analysis of biological samples: With the emergence of POC microfluidic platforms for single-cell analysis [34], new techniques are needed to efficiently classify cells and conduct biochemical experiments on multiple cell types concurrently [35].

The realization of a digital-microfluidic platform that supports the above characteristics requires a new synthesis paradigm that is based on the realistic modeling of biomolecular protocols. For example, a control-flow-aware synthesis capability is needed to determine which pathway must be followed to achieve successful completion of a bioassay based upon the monitored status of the assay during run time. Hence, this paper is focused on: 1) a biochip architecture and optimization methods to tackle the challenges associated with biomolecular protocols; and 2) simulation and evaluation of the biochip architecture and optimization solutions, based on realistic scenarios extracted from benchtop implementation of biomolecular protocols. Such algorithmic innovations can fill the gap between control/monitoring in the physical space and online biochemistry-on-chip synthesis in the cyber space, and therefore promote the dissemination of DMF technology in translational and clinical research.

### B. Organization of the Paper

The rest of this paper is organized as follows. Section II describes an overview of integrated sensing techniques in cyber-physical DMFBs and a review of previous design-automation and optimization solutions. In Section III, we describe three design-automation challenges for quantitative-analysis protocols. Details about the design-automation support and system evaluation for these challenges are provided in Sections IV–Sections VI. Finally, conclusions are drawn in Section VII.

<sup>1</sup>Detailed explanation of a quantitative-analysis protocol can be found in [32].

## II. INTEGRATED SENSING AND PRIOR DESIGN-AUTOMATION SOLUTIONS

In this section, we review miniaturized sensing technologies, which enable cyber-physical integration in DMFBs. We also review previous design-automation methods.

### A. Sensing Systems

Sensing systems on DMFBs can be used to detect a biological event or quantify the existence of a constituent biochemical analyte during an analytical procedure. Examples of biochemical analytes are toxins in food, pollutants in air/water, and intracellular biomolecules. Researchers have developed numerous integrated sensing techniques that are used in concert with DMFBs; these techniques can be classified into three main categories [36]: 1) optical detection (e.g., absorbance, fluorescence, chemiluminescence, and surface plasmon resonance detection); 2) mass spectrometry; and 3) electrochemical detection. Fluorescence detection is the dominant sensing method for biomolecular analysis—it is widely used because the fluorescent labeling techniques are well established and they provide quantification results with superior selectivity and sensitivity despite small sample volumes [18].

1) *Fluorescence Detection*: Contemporary biomolecular research relies extensively on analyzing amplified DNA signals in correspondence with biological events such as chromatin packing alteration [37]. To develop a single-chip solution that incorporates the purification and detection of DNA signals, a fluorescence sensor along with a fluorescent dye injected to the sample is used to analyze the fluorescence generated by the on-chip quantitative polymerase chain reaction (qPCR) protocol [34]. In [25], a custom-built fluorescence sensor is reported for qPCR experiments.

A key advantage of fluorescence sensors is that they can be used to visualize multiple fluorescent molecules by simply adjusting the excitation and emission filters to detect the fluorophores in a sample—this feature enables multiplex genomic analysis even with multiple samples.

2) *Lensless Sensing*: Although optical microscopy provides high-resolution imaging of biomolecular samples, it relies on high-magnification and high-numerical-aperture lenses. A new trend has recently emerged to design lensless microscopy that can provide high-resolution images without the use of any focusing lenses, offering cost-effectiveness and portability features [38], [39]. Such techniques can be easily used to detect and monitor cells, and they can be widely utilized in the early phases of quantitative-analysis workflow.

### B. Design-Automation Methods

Design-automation research for DMFBs has become increasingly important over the past decade. Initially, a top-down design methodology was introduced, allowing

researchers to work on three different levels of synthesis: architectural level, physical level, and chip level [40]. Architectural-level synthesis focused on scheduling and resource binding of biochemical operations [41]–[43]. In physical-level synthesis, methods have been developed for on-chip droplet routing and cross-contamination avoidance while routing [44]–[47], whereas chip-level synthesis included mapping of control pins to electrodes and pin-count minimization [48], [49].

Despite the advances brought by these methods [50], they are only limited to a simple setting where the sequence of biochemical operations is specified in advance. Therefore, these early methods cannot support bioassay uncertainties, which may arise due to error occurrences or conditional sample processing that is typical for biomolecular protocols. In other words, support for cyber-physical system design was not considered in these early methods, and they ignored the interplay between hardware and software in the DMFB.

Several on-chip sensing techniques have been developed, providing an opportunity for designing cyber-physical microfluidic biochip [51]. In such a biochip, the physical space is formed by the sensor connectivity, which ensures real-time detection of bioassay droplets. The cyber space, on the other hand, is represented by online control software that captures the detection outcome, received from the physical space, and enforces a suitable action through biochip actuators. Design and optimization techniques for cyber-physical DMFBs have considered error recovery [51]–[55] and termination control of a biochemical procedure such as PCR [56].

A drawback of these cyber-physical designs is that they do not support real-time decision making for biomolecular protocols. In such protocols, multiple samples are involved in biochemical reactions, with each sample pathway being independent of the others. For example, in [57], Gao *et al.* developed a complex control mechanism that can be used for droplet volume measurement and position control on the chip array. However, due to design complexity, the control of multiple droplets using this mechanism is impractical. Similarly, in [58], a hardware-based cyber-physical framework has been designed to support error recovery in DMFBs. This framework provides a practical demonstration for reliable bioassay execution, but it lacks support for real-time biomolecular measurement.

Therefore, a major limitation of all previous work on design automation and cyber-physical system integration is that they only address challenges associated with simple droplet manipulation on a biochip. Hence, a new design methodology is necessary to transition DMFBs towards realistic implementation of actual biomolecular protocols.

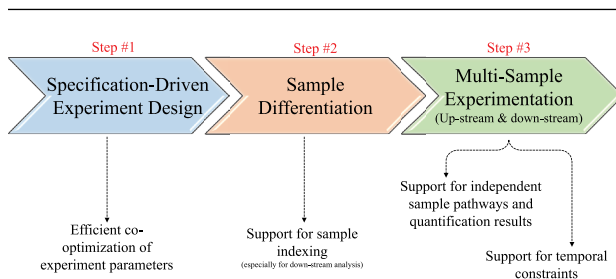
## III. DESIGN-AUTOMATION METHODOLOGY FOR QUANTITATIVE ANALYSIS

An experimental framework for quantitative-analysis studies must be designed with the following key objectives in mind.



- 1) An experimental framework needs to be optimized to utilize the smallest amount of samples to obtain answers to questions in biology with the highest possible precision. An example of this challenge is the realization of the ChIP-seq protocol, wherein the reduction of the number of cells is needed to ensure that ChIP-seq can be widely adopted [59]. However, since sample reduction in turn entails a significant decrease in the precision of reported DNA-protein bindings (i.e., sequencing coverage), co-optimization of framework design and sequencing coverage is essential.
- 2) From a biology perspective, an experimental framework is developed for one of two objectives: a) identification of biomolecules—also known as upstream analysis—that exhibit specific or abnormal biological behaviour (e.g., fluorescence-based gene expression/suppression due to enzymatic reaction); and b) causative exploration of biomolecules—also known as downstream analysis—that contribute to the *in vivo* interactions leading to such biological behaviour (e.g., the impact of protein binding to DNA at specific genomic regions). Such causative exploration requires a method for indexing/barcoding samples at the end of up-stream analysis to account for cellular heterogeneity [60].

As a result, the workflow of biomolecular analysis that needs to be adopted consists of three main stages: specification-driven experiment design, sample labeling and differentiation, and multisample experimentation, as shown in Fig. 4. The first stage, i.e., experimental design, is carried out before experimentation and it seeks co-optimization of experiment goals (e.g., amount of input sample, resource, and sequencing coverage) [61]. A decision at this stage impacts the protocol procedure. The second stage, i.e., sample labeling and differentiation, is focused on developing scalable methods for differentiation of heterogeneous samples within downstream analysis [62]. The third stage, i.e., multisample experimentation, represents the implementation of protocol procedures using multiple sample pathways in order to obtain quantitative results; this stage applies to both upstream and downstream analyses. In this paper, we



**Fig. 4. The workflow of biomolecular quantitative analysis and the associated design-automation challenges.**

focus only on the design-automation challenges associated with the second and third stages.

1) *Independent sample Pathways, Online Decision Making, and Quantification of Results:* Quantitative results are reported numerically and are compared against an accompanying reference interval for interpretation. For this purpose, multiple samples are treated through independent sample pathways. Subsequently, the distributed detection results of the sample pathways are collected and fused to report a final quantitative result. Therefore, to support multiple sample pathways, it is necessary to extend the traditional sequencing graph model [see Fig. 3(a)] to represent the associated bioassay protocols for all samples. Note that the procedures associated with these protocols can either be identical for all samples or be unique to each sample type according to the application specifications. This modeling technique is similar to the multifunctional design scheme presented in [63], which combines a set of multiplexed bioassay protocols in the same platform. Our solution, however, executes the bioassay protocols concurrently and collects detection results from all these protocols (using firmware) to reach a meaningful conclusion.

Another important feature of quantitative-analysis protocols, such as epigenetic protocols, is the inherent uncertainty about the sequence of fluidic steps for each sample. In other words, based on our recent benchtop experience with multiple sample pathways, it is necessary to incorporate decision-making capability into our microfluidic framework. For example, nucleic-acid isolation during the execution of qPCR-based gene-expression analysis must be followed with a decision on whether the quality of the isolated NA allows subsequent cDNA synthesis or requires further steps to get rid of the residual protein and debris—such “if-then-else” reconfiguration can be performed either automatically using cyber-physical integration or using human-in-the-loop if necessary.

2) *Support for Temporal Constraints:* Temporal constraints may arise during protocol execution due to physical phenomena such as droplet evaporation or deadlines imposed by the target chemistry, e.g., degradation of samples and reagents [33]. Therefore, we need to provide a systematic approach that handles such constraints during the course of an experiment in order to ensure robust and reproducible quantitative results. An example of an upper-bound temporal constraint can be seen in ChIP protocol, in which the protein is initially cross-linked to DNA. It has been shown that an excessively long cross-linking time results in the majority of chromatin being resistant to shearing, thus degrading chromatin shearing process [64]. Therefore, an upper-bound temporal constraint on cross-linking reactions must be specified; this temporal constraint is often dependent on the sample type.

3) *Support for Indexing/Labeling Samples:* Microfluidic solutions have been developed to classify cell types (using cell sorting) or perform biochemical analysis at the cell level on preisolated types of cells [65]. With the emergence

of POC microfluidic platforms for single-cell analysis, new techniques are needed to efficiently classify cells and conduct biochemical experiments on multiple cell types concurrently. System integration and design automation are major challenges in this context. To enable such an integrated system, it is necessary to develop a microfluidic platform and a design-automation framework that combine the following steps of the single-cell experimental flow: a) cell encapsulation and differentiation; b) sample-indexing (bar-coding) to keep track of the samples identity; and c) type-driven biochemical analysis.

We discuss design-automation solutions associated with the above challenges in Sections IV–VI, respectively. Real-world microbiology applications are used to motivate these solutions.

#### IV. SYNTHESIS FOR MULTIPLE SAMPLE PATHWAYS: REALIZATION OF REAL-TIME GENE-EXPRESSION ANALYSIS

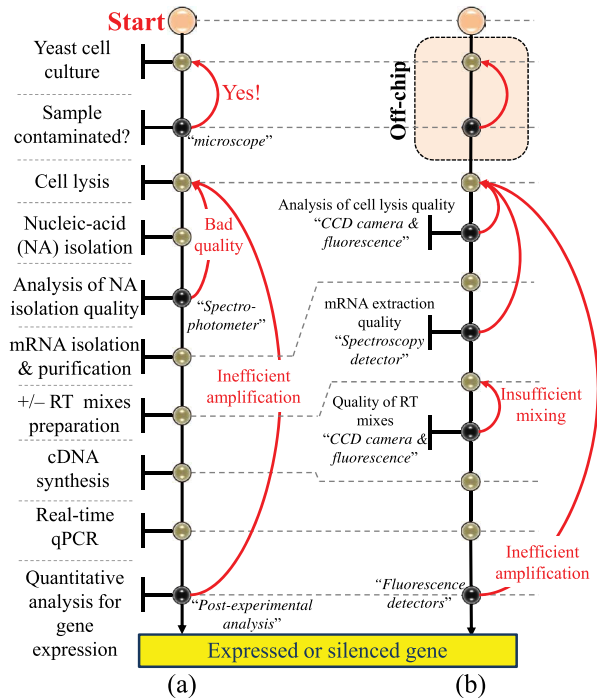
In this section, we present a design and optimization framework to control multiple sample pathways in quantitative-analysis protocols. This basic solution can be used to solve the synthesis problem for downstream protocols, which use samples that were processed and differentiated at an earlier stage; see Fig. 4. The integration of this framework with a sample-differentiation utility is introduced in Section VI.

We describe an algorithmic model for a protocol that studies the gene-expression levels in molecular biology [31]. The design-automation framework adopts a physical-aware reconfiguration scheme to coordinate resource sharing among multiple sample pathways. Resource-sharing constraints are incorporated into the design-automation flow to ensure reliable execution of gene-expression analysis.

This solution is not only applicable to the gene-expression analysis protocol, but it is also amenable to other forms of biological quantification such as concentration (e.g., in glucose testing) [66]. Gene-expression analysis is considered as an example in this section due to its widespread use in biomolecular analysis.

##### A. Miniaturization of Gene-Expression Analysis

We leveraged our recent benchtop experience with the study of gene-expression analysis to pinpoint the key characteristics of a sample pathway [31]. As shown in Fig. 5(a) [67], benchtop realization of biomolecular analysis involves situations where a sample droplet can be processed in an unpredictable manner by various biochemical reactions, e.g., cell lysis, and therefore real-time decision making takes place within the sample pathway. Hence, we have developed an enhanced and miniaturized protocol, shown in Fig. 5(b), where the protocol procedure includes



**Fig. 5. A protocol flowchart for gene-expression analysis using: (a) a benchtop setup; and (b) DMFBs.**

the following bioassay implementations: 1) lysing of cultured cells to obtain intracellular materials such as DNA and mRNA; 2) isolation of mRNA using magnetic beads, enzymes, and a washing procedure; 3) reverse transcription (RT) of mRNA into the corresponding complementary DNA (cDNA) using gene-specific RT kit; and 4) thermal cycling of cDNA via qPCR to amplify the target gene.

Similar to the benchtop approach, on-chip realization of gene-expression analysis requires the execution of this protocol on multiple sample droplets. Therefore, an autonomous DMFB for gene-expression analysis must support concurrent manipulation of independent samples. Using the decision points shown in Fig. 5(b), sample-dependent decision making is incorporated, via cyber-physical system integration, to overcome the inherent unpredictability of the order of bioassays. Furthermore, the specification of the protocol efficiency and the level of gene expression are included on-chip in the feedback system. Details on determining the protocol efficiency and gene-expression level can be found in [31].

There are major advantages of using microfluidic platforms compared to benchtop (conventional) setup for gene-expression analysis. For example, it has been reported in [68] that PCR execution using a DMFB is three times faster than using a conventional setting, and it also offers an inexpensive miniaturized form of the PCR assay. In addition, microfluidic platforms are versatile, portable, and amenable to POC use, and therefore they can be used in the hospital or

outpatient setting. Such characteristics and others motivate the use of microfluidics for microbiology applications such as gene-expression analysis.

In embedded systems, handling control dependencies among tasks to support physical-aware reconfiguration is a traditional hardware–software codesign problem that is modeled using conditional process graphs [69], [70]. Similarly, in DMFBs, reconfiguration of multiple sample pathways is captured using a control-flow graph (CFG)  $G_c = (V_c, E_c)$  that represents the biomolecular protocol. In this graph model, every node  $v_c \in V_c$  (referred to in this context as a supernode) represents a bioassay, e.g., mRNA extraction, and a directed edge  $e_c(v_{c1}, v_{c2}) \in E_c$  indicates a potential transition (i.e., decision making) path from  $v_{c1}$  to  $v_{c2}$ . The synthesis framework synthesizes the supernodes based on the decisions made during protocol execution.

We also model the fluid-handling operations corresponding to a supernode  $v_c$  using a directed sequencing graph (DAG)  $G_s = (V_s, E_s)$ , where the set of nodes  $V_s$  represents the types and timing of the operations, and the set of edges  $E_s$  shows the interdependencies among these operations. Fig. 6 illustrates the representation of the protocol used for gene-expression analysis.

A new design-automation methodology is required to put this model into practical use. Our premise is that dynamic resource allocation and spatial reconfiguration are essential design-automation methods due to the following reasons: 1) biomolecular analysis is performed using multiple sample pathways that are manipulated concurrently and investigated independently; and 2) there is a need for a robust DMFB design, because a nonrobust design may lead to system failure or inaccurate quantification results. Hence, we adopt dynamic resource allocation on DMFB to handle a collection of sample pathways and use a spatial

reconfiguration technique to enforce reliability constraints. Utilizing both mechanisms in a combined manner ensures robust miniaturization of a microbiology application on a resource-limited DMFB.

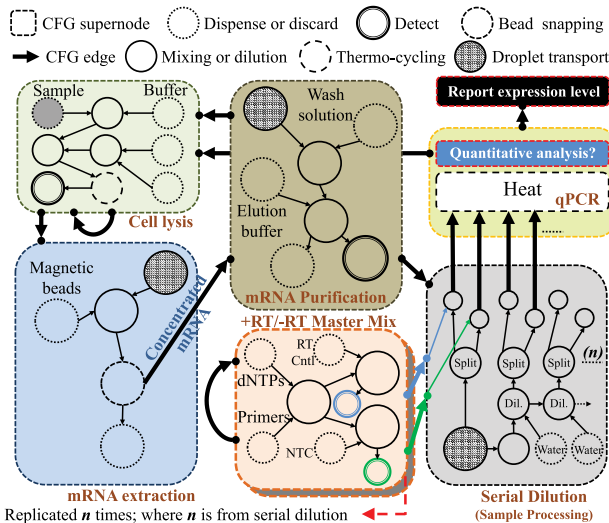
## B. Spatial Reconfiguration

As described in Section IV-A, miniaturization of gene-expression analysis using DMFBs is realized through concurrent manipulation of multiple sample pathways, yet the complete sequence of bioassay implementations for each pathway cannot be specified in advance. As a consequence, the allocation of biochip modules to protocol samples can only be employed via a reconfiguration technique that can correlate specifications of the bioassays with the available on-chip devices—this reconfiguration scheme is referred to as spatial reconfiguration.

Since multiple samples share a limited set of resources, the above reconfiguration scheme must consider the degradation caused by a bioassay. An electrode can exhibit a short lifetime if it is excessively used by the protocol samples. Based on [71], an electrode’s lifetime, characterized by the threshold voltage needed for actuation, can be divided into three phases: reliable operation, safety margin, and breakdown. In the reliable operation phase, the threshold voltage remains constant. When an electrode is transitioned to the safety margin phase, the threshold voltage increases linearly. However, in the breakdown phase, a significant increase in the threshold voltage is required to transport a droplet. This increase in the electrowetting voltage, in turn, quickly damages the electrode as it causes dielectric breakdown [72].

The degradation model described above can be utilized to regulate resource allocation among bioassays. For this purpose, we analyze the completion time and the expected electrode degradation for the critical resources, e.g., heaters, over a varying range of resource availability. Hence, we specify three different schemes of spatial reconfigurations to regulate resource allocation.

- 1) Nonreconfigurable scheme: In this scheme, on-chip devices are allocated *a priori* to protocol bioassays [34], [71]. The realization of this scheme is straightforward, and it is, therefore, adopted by current prototypes.
- 2) Restricted resource sharing: Restriction is applied to limit the reconfigurability of the shared devices among bioassays. For example, a heater can be shared between qPCR and cell lysis for thermal manipulation or sample processing, but it cannot be used by other bioassays to limit the degradation of the associated electrodes. It is apparent that this scheme achieves higher completion time.
- 3) Unrestricted resource sharing: In this scheme, no restriction is imposed on resource sharing. Therefore, heaters cannot only be used for thermal cycling or



**Fig. 6. A CFG representation of the gene-expression analysis protocol for a single sample pathway [67].**

sample processing in cell lysis or qPCR, but they can also be used for sample processing in all bioassays to reduce the total completion time. A drawback of this scheme is that it reduces chip lifetime.

Our goal is to combine the advantages of both restricted resource sharing and unrestricted resource sharing, i.e., to achieve lower completion time and less degradation of the chip, in a single resource-allocation mechanism. We refer to this mechanism as degradation-aware resource allocation, and it is described next.

### C. Shared-Resource Allocation

In this section, we describe the resource-allocation problem and introduce the algorithmic solution.

1) *Problem Formulation*: The notation used to describe resource allocation is listed in Table 1. The DMFB considered here is composed of three types of resources: a) nonreconfigurable resources such as input and output ports; b) sample-processing resources such as mixers; and c) reconfigurable resources such as heaters, optical detectors, and magnetic wires. The reconfigurable resources can be shared among the protocol's bioassays, whereas the allocation of a nonreconfigurable or a samples-processing resource is limited to a specific bioassay. Access control to biochips resources is regulated as follows.

- Each bioassay  $b_i \in B$  is mapped to a space of nonreconfigurable and sample processing modules.
- A shared module  $r \in R_{sh}$  is essential for a bioassay  $b_i$  if and only if the absence of  $r$  leads to a failure in execution of  $b_i$ . An example of this relation is the heater resource for a thermal-cycling bioassay.
- A shared module  $r \in R_{sh}$  that is granted to a bioassay  $b_i$  may not be essential for the execution of the bioassay, i.e.,  $r_i \subset gr_i$ . Typically, this module can be used for sample processing.

Based on the above discussion, we introduce the problem formulation as follows.

**Inputs:** 1) The digital microfluidic library, which describes the types, locations, and operation timing of the on-chip modules.

**Table 1** Notations Used in Section IV

$B$	The complete set of bioassays
$b_i$	A bioassay $b_i$
$R$	The complete set of the chip devices
$R_{sh}$	The set of shared, reconfigurable modules
$r_i$	The set of shared resources that are essential for a bioassay $b_i$
$gr_j$	The set of shared resources that are granted to a bioassay $b_j$
$D_i^j$	The degradation caused by a bioassay $b_i$ on the shared resources $gr_j$
$T_i^j$	The completion time for a bioassay $b_i$ when the resources $gr_j$ are used

2) The protocol CFG  $G_c = \{V, E\}$ , where  $V = \{V_1, V_2, \dots, V_m\}$  represents the supernodes of  $m$  bioassays and  $E = \{(V_i, V_j) \mid 1 \leq i, j \leq m\}$  signifies biological dependencies between all pairs of bioassays  $b_i$  and  $b_j$ . A supernode  $V_i$ , in turn, encapsulates a DAG  $G_s = \{V_s, E_s\}$ , where  $V_s = \{V_{s_1}, V_{s_2}, \dots, V_{s_n}\}$  represents  $n$  bioassay operations and  $E_s = \{(V_{s_i}, V_{s_j}) \mid 1 \leq s_i, s_j \leq n\}$  represents dependencies between all pairs of biochemical operations  $s_i$  and  $s_j$  that belong to the bioassay  $V_s$ .

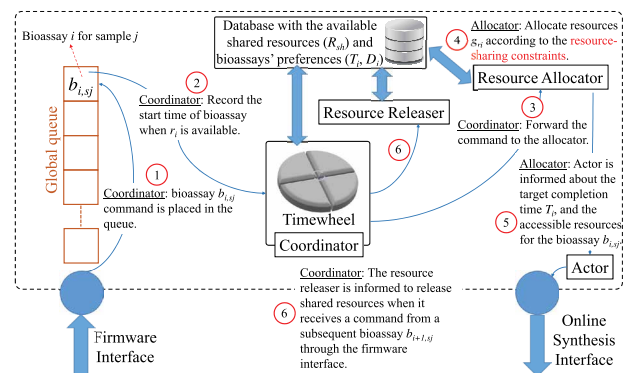
3) The initial resource requirement for each bioassay  $b_i$  (i.e., the resource preferences). Moreover, the values of the bioassay's completion time  $T_i^j$  and the degradation level  $D_i^j$  as a function of the granted resources  $gr_j$  are also given.

4) The resource-allocation constraints described in Section IV-C2.

**Output:** Allocation of chip resources to the bioassays such that the constraints imposed on resource allocation (e.g., degradation constraints) are satisfied.

2) *Resource-Allocation Constraints and Algorithm*: Fig. 7 depicts the components and the sequence of actions (highlighted by the numbers 1–6) of the resource-allocation mechanism. A timewheel is utilized to keep track of resource utilization and it is managed by the coordinator. First, when a decision is made by the firmware for a certain pathway, the firmware translates this decision into a bioassay command that is communicated to the coordinator; a global queue is used by the coordinator to store all the received commands. Next, when the command is ready to be processed, the coordinator informs the resource allocator agent with this command. In response, the resource allocator performs two tasks: a) it matches the preferences of the bioassay with the available resources; b) it enforces scheme-specific constraints on resource allocation. Finally, a solution for resource allocation is obtained and forwarded to the actor, which triggers the online synthesis process.

If the resource allocator adopts restricted resource sharing, then it ensures that the requesting bioassay can execute



**Fig. 7.** The components of the shared-resource allocator [31].



using shared resources that are only essential for the correctness of this bioassays; i.e.,  $r_i = gr_j$ . Hence, the worst case computational complexity for allocating resources to a bioassay  $b_i$  is  $O(|r_i|)$ . On the other hand, the resource allocator in unrestricted resource sharing allows nonessential shared resources to be utilized by any requesting bioassay. In this case, the worst case computational complexity for allocating resources to a bioassay  $b_i$  is  $O(|R_{sh}|)$ .

By using the third scheme, namely degradation-aware resource allocation, the degradation level at a certain shared resource is instantaneously updated whenever it is being used to execute a bioassay. As a result, the restriction on sharing this particular resource can change over the course of protocol execution. In other words, a shared resource can initially be allocated to any requesting bioassay without restrictions. Afterwards, when the reliable operation time for the electrodes at this resource is exceeded, restriction is imposed by the resource allocator to limit the access to this particular resource. Note that since electrode degradation is considered in resource allocation, resources are sorted according to their degradation level whenever a bioassay  $b_i$  requires resource allocation. Therefore, the worst case computational complexity of this scheme is  $O(|R_{sh}|^2)$ .

Motivated by the literature on real-time embedded systems [73], [74], we foresee many other research opportunities for optimizing resource allocation in DMFBs. Objectives such as fault tolerance and experiment-cost reduction are likely to be the main focus of these efforts.

#### D. Simulation Results

The above schemes have been developed using C++ and reported in [67]. The resource preferences for protocol bioassays were specified using offline synthesis simulations [48].

We used two biochip designs to simulate gene-expression analysis; these designs are labeled  $C_{NR}$  (nonreconfigurable chip) and  $C_R$  (reconfigurable chip). For reliable dispensing in both designs, a specific interfacing port is used for each liquid. Unlike sample processing (SP) modules, reconfigurable modules, such as heaters (H), CCD camera region (CCD), detectors (OD), and magnetic devices (M), are located in the shared area of the reconfigurable chip, i.e.,  $C_R$ . The sizes of electrode array for both  $C_{NR}$  and  $C_R$  are  $18 \times 18$  and  $17 \times 17$ , respectively.

We evaluate four resource-allocation schemes: 1) nonreconfigurable scheme (NON); 2) restricted resource sharing (RR); 3) unrestricted resource sharing (NR); and 4) adaptive degradation-aware resource sharing (DA). By using the protocol described in Section IV-A, our evaluation is based on simulation scenarios where three samples are being concurrently manipulated; these samples are as follows:  $S_1$  (GFP gene-targeted sample),  $S_2$  (YFP gene-targeted sample), and  $S_3$  (actin gene-targeted sample).

Table 2 summarizes the bioassay notation and the resource requirements for each bioassay. Our evaluation comprises three different cases in terms of the length of

**Table 2** Bioassay Notation and Resource Requirement

Bioassay	Notation	Minimum resource requirement
Cell Lysis	CL	SP, H, and CCD
mRNA Extraction	mE	SP and M
mRNA Purification	mP	SP and OD
RT Master Mix	MM	SP and OD
Serial Dilution	SD	SP
Thermal Cycling	TC	H

sample pathways; these simulation cases are as follows.

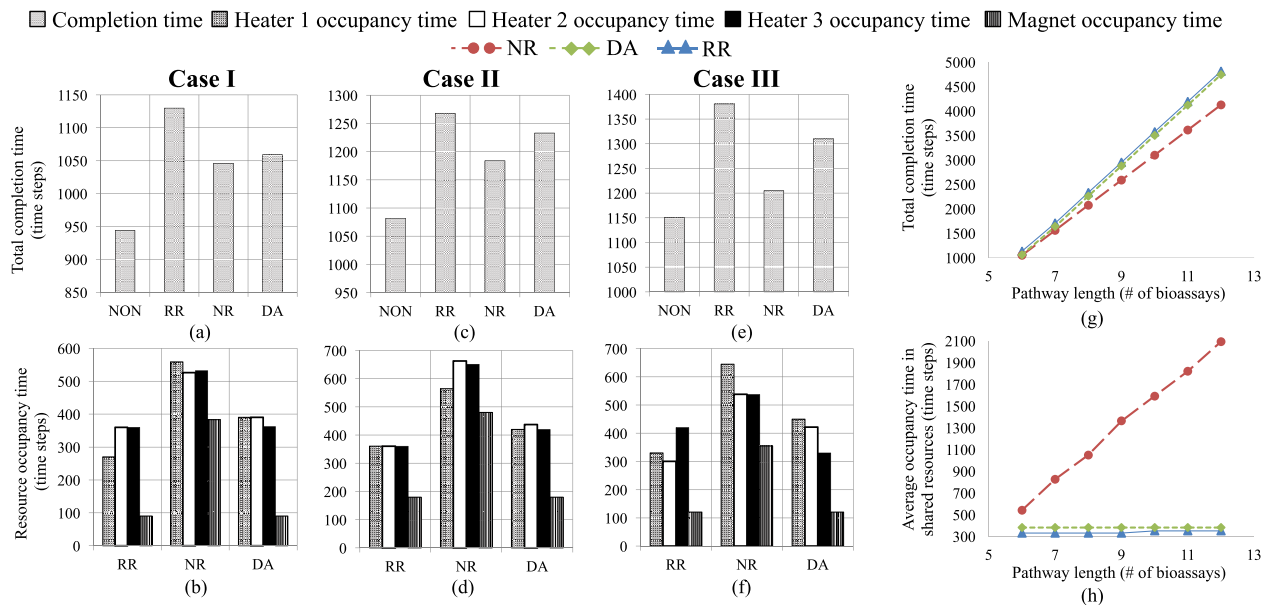
1) Short homogeneous pathways (Case I): an optimistic case in which all the three samples follow the same shortest pathway (CL-mE-mP-MM-SD-TC). 2) Long homogeneous pathways (Case II): a pessimistic case in which all the three samples follow the same long pathway (CL-mE-mP-CL-mE-mP-MM-SD-TC). 3) Heterogeneous pathways (Case III): a realistic case in which the pathways of these samples are different. The considered pathways are as follows: CL-mE-mP-MM-SD-TC, CL-mE-mP-CL-mE-mP-MM-SD-TC, and CL-CL-mE-mP-MM-MM-SD-TC, respectively.

Comparison between the resource-allocation schemes is performed using two metrics: 1) protocol completion time (in time steps); and 2) the occupancy time for the shared resources (heaters and magnet modules) to report the degradation level. For fair comparison between nonreconfigurable and resource-sharing schemes, the chip designs used with the nonreconfigurable scheme (i.e.,  $C_{NR}$ ) is designed such that the resources are sufficient to process only a single pathway; therefore, the resources are not replicated with the number of samples. In addition, due to the absence of resource sharing in the nonreconfigurable scheme, the occupancy time is not considered in our analysis. The biochip specifications in both reconfigurable ( $C_R$ ) and nonreconfigurable ( $C_{NR}$ ) settings are shown in Table 3.

1) Case I: Short Homogeneous Pathways: This case arises when all the samples are perfectly grown in a well-controlled medium. In addition, the reagents are contamination free. We compare the four methods in terms of completion times; see Fig. 8(a). As expected, nonreconfigurable resource allocation leads to the shortest completion time. Restricted resource sharing, on the other hand, shows the worst completion time. We also note that we can use adaptive, degradation-aware resource allocation (DA) to achieve a short completion time, while the chip resources are not

**Table 3** Number of On-Chip Modules and Array Electrodes in  $C_{NR}$  and  $C_R$

Resource	# Modules		# Electrodes	
	$C_{NR}$	$C_R$	$C_{NR}$	$C_R$
SP	1 (CL), 2 (mE), 2 (mP), 2 (MM), 4 (SD)		176	176
H	2 (CL), 3 (TC)	3	80	48
M	1 (mE)	1	18	18
OD	2 (mP), 2 (MM)	2	36	18
CCD	1 (CL)	1	9	9
Chip size	22	18	$18 \times 18$	$17 \times 17$



**Fig. 8. (a)–(f) Comparison between the four resource-allocation schemes—NON, RR, NR, and DA—executing three pathways. (g)–(h) Performance of the shared resource-allocation schemes with various lengths of homogeneous pathways.**

severely degraded (degradation is measured in terms of the occupancy time); see Fig. 8(b).

- 2) Case II: Long Homogeneous Pathways: We next study the case where all the samples are resuspended for additional bioassays. As shown in Fig. 8(c) and (d), the proposed schemes show the same profiles of completion times and degradation levels as in Case I, but with higher values. However, the completion time for DA is closer to that obtained with restricted resource sharing.
- 3) Case III: Heterogeneous Pathways: This is a realistic case that emerges due to the inherent uncertainty about the biological contents of each sample. It introduces the challenge of deciding the allocation of resources among the heterogeneous pathways at runtime. Again, we evaluate the allocation schemes based on the completion time [Fig. 8(e)] and the degradation level [Fig. 8(f)].
- 4) Convergence of Degradation-Aware Resource Allocation: Finally, we study the convergence of the DA method. We analyze the completion time and the average degradation level of the shared resource-allocation schemes with various lengths of homogeneous pathways, as shown in Fig. 8(g)–(h). We observe that the completion time for DA begins to converge to its counterpart in RR when the length of the pathway is increased. However, due to the restrictions imposed by both RR and DA on resource allocation, the degradation levels remain below a certain limit even when we increase the pathway length substantially, as shown in Fig. 8(h). In real-life

scenarios, chip users tend to make optimizations in order to make sure that the protocol is finished as early as possible to avoid droplet evaporation [33]. These scenarios make the DA method especially attractive in practice.

## V. SYNTHESIS FOR LARGE-SCALE PROTOCOLS WITH TEMPORAL CONSTRAINTS: REALIZATION OF REAL-TIME EPIGENETIC ANALYSIS

In Section IV, a synthesis methodology was introduced to support nontrivial biology-on-a-chip applications at the downstream phase (e.g., quantitative gene-expression analysis). However, a drawback of the previous method is that it makes spatial reconfiguration decisions at the bioassay level (i.e., “locally”), and it does not capture interactions between multiple sample pathways at the protocol level. In other words, resource sharing among different sample pathways is achieved at a coarse-grained level, thus biochip devices are not efficiently exploited, particularly when further complicated protocols such as gene-regulation analysis (referred to here as epigenetics) are considered. Furthermore, this work does not consider the upper-bound temporal constraints imposed by the application domain; these constraints may arise due to physical phenomena such as droplet evaporation and deadlines imposed by the target chemistry, e.g., degradation of samples and reagents. It was found that droplets that are subjected to overly long-lasting reactions (especially in a medium of air) are susceptible to evaporation, which impacts the efficiency of enzymatic reactions and alters protocol outcomes [33].

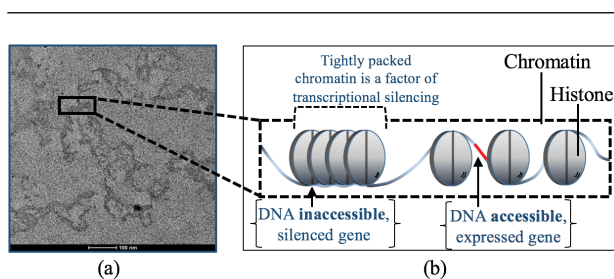
In this section, we overcome the above drawbacks by presenting a system design and a design-automation method that carries out task assignment and scheduling for cyber-physical DMFBs for quantitative analysis, e.g., the study of alterations in gene expression or cellular phenotype of epigenetics [75]. The proposed method scales efficiently to multiple independent biological samples and supports on-the-fly adaptation under temporal and spatial constraints. Synthesis of epigenetics is mapped to real-time multiprocessor scheduling and formulated as an integer programming problem. Since the scheduling problem is NP-hard, we develop a heuristic for dynamic fluidic task scheduling to respond to protocol-flow decisions. The proposed algorithm provides resource sharing and handles droplet evaporation.

Furthermore, to promote component-based design [76] in DMFBs, the interaction between the proposed algorithm and other system components (actuation and firmware) is demonstrated using an embedded microcontroller board.

### A. Miniaturization of Epigenetic-Regulation Analysis

One of the important uses of gene-expression analysis (GEA) described in the previous section [flowcharts corresponding to the benchtop protocol and the miniaturized implementation are shown in Fig. 5(a) and (b), respectively] is in epigenetics, which identifies changes in the regulation of gene expression that are not dependent on gene sequence. Often, these changes occur in response to the way the gene is packaged into chromatin in the nucleus. For example, a gene can be unfolded (“expressed”), be completely condensed (“silenced”), or be somewhere in between. Each distinct state is characterized by chromatin modifications that affect gene behavior. An improved understanding of the *in vivo* cellular and molecular pathways that govern epigenetic changes is needed to define how this process alters gene function and contributes to human disease [37].

Based on our benchtop study of gene-expression analysis [31], we assessed the relevance of chromatin structure on regulation of the gene function. A second benchtop experiment was carried out to image yeast chromatin samples under a transmission-electron microscope (TEM). Fig. 9 relates the outcome of the experiment to gene-regulation behavior based on chromatin structure. With multiple

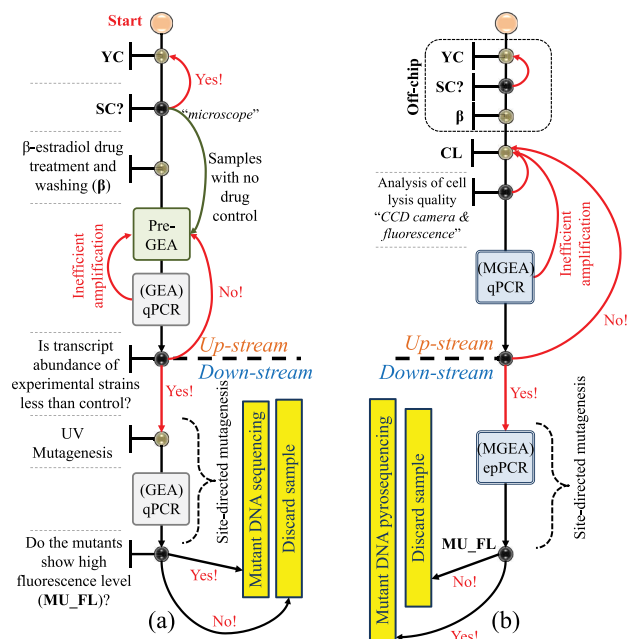


**Fig. 9. (a) TEM image of chromatin. (b) Correlation with chromatic control of epigenetic gene regulation.**

samples and with several causative factors affecting chromatin behavior, implementing epigenetic-regulation analysis using a benchtop setting is tedious and error prone; thus this preliminary benchtop study motivates the need to miniaturize epigenetic-regulation analysis. The benchtop study also provides important guidance on the design of the miniaturized protocol for a DMFB [75]. Fig. 10(a) depicts a flowchart of the benchtop protocol. The protocol consists of two stages.

- 1) Upstream stage in which the transcriptional profile (gene expression) of a GFP reporter gene is investigated. Control samples (GFP not under epigenetic/drug control) and experimental strains (under epigenetic control) were analyzed by qPCR. The goal was to explore how chromatin-folding alterations influence gene expression.
- 2) Downstream stage in which novel modifiers of epigenetic gene regulation are identified. Samples are mutagenized, for example by ultraviolet radiation, and cells whose transcriptional activity has been enhanced or suppressed are analyzed further. Following quantitative gene expression analysis, the causative mutation can be identified by whole genome sequencing. The role of the genes in epigenetic processes can be verified by additional studies, including TEM analysis, ChIP, and other assays.

Similar to Section IV, reliable concurrent manipulation of independent samples requires the incorporation of sample-dependent decision making into the protocol. We



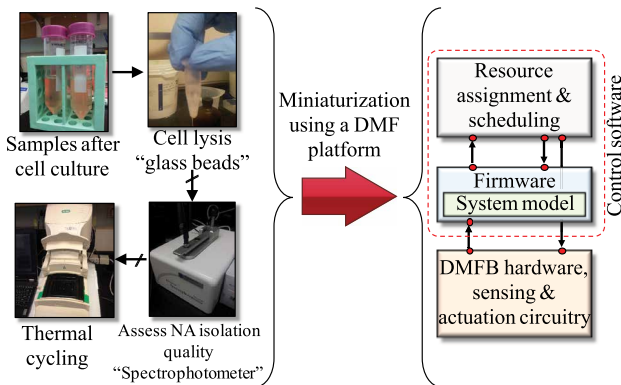
**Fig. 10. Protocol for epigenetic gene-regulation analysis using (a) benchtop setup; and (b) DMFBs.**

have developed a miniaturized protocol for epigenetic-regulation analysis; see Fig. 10(b). Using DMFBs, both upstream and downstream stages are carried out using the protocol for gene-expression analysis, followed by DNA pyrosequencing [20], to identify the sequences of the generated mutations. To provide a successful transformation of the complex epigenetic protocol into a biochip setting, a layered structure of a digital-microfluidic system is deployed.

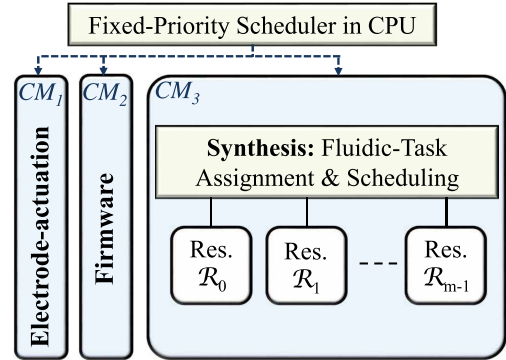
Fig. 11 illustrates the components required for on-chip implementation of the protocol, and the interactions among them. The control software consists of a system model, embedded in a firmware layer, and a real-time resource assignment and scheduling layer. The protocol is initially synthesized to process the fluids across independent sample pathways. At the end of a bioassay, a sample is subjected to a detection operation that triggers the start of a decision-making process. The firmware receives the sensor readout, analyzes the data, and produces a decision to the scheduler. Real-time synthesis is then employed to provide the needed actuation sequences to adapt the system to the new situation.

In principle, real-time synthesis is sufficient to capture the dynamics of fluid-handling operations within multiple sample pathways. However, an integrated system for epigenetics must take into account the impact of other active components: electrode actuation and firmware. In other words, the system scheduler is responsible for the timely coordination between the periodic loading of actuation sequences ( $CM_1$ ), firmware computation ( $CM_2$ ), and synthesis execution ( $CM_3$ ).

Therefore, we represent our DMF platform as a hierarchy of components (Fig. 12);  $CM_1$  triggers a set of periodic tasks to stimulate the CPU to transfer actuation sequences from the controller memory to the biochip control pins at the start of every actuation period. The tasks triggered by  $CM_2$  are invoked sporadically at every decision point within the protocol. Note that the durations of tasks generated by  $CM_1$  and  $CM_2$  are fixed and can be easily determined using offline simulation.



**Fig. 11. Proposed layered structure of a DMF platform to miniaturize a quantitative protocol. In this section, we design the system model and the real-time scheduler.**



**Fig. 12. Hierarchical scheduling of DMF system components.**

Our focus in this section is on the control-software design and optimization, specifically, system modeling and real-time scheduling. In Section V-D3, we present a demo using a microcontroller and simulation data. In Sections V-B and Sections V-C, we present details about the system model and real-time scheduling.

## B. System Model

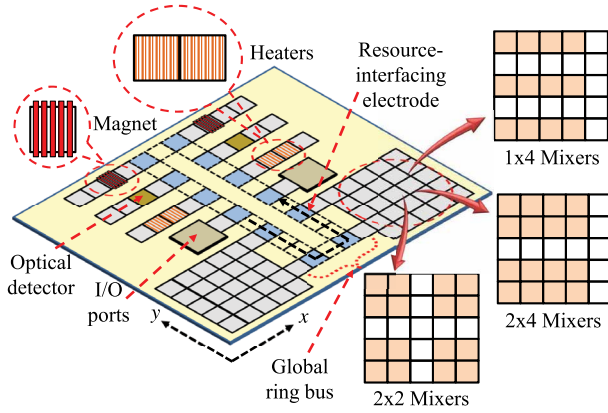
We model the DMF system for gene-regulation analysis in terms of real-time computing systems.

1) *Biochip Resources*: The DMF platform includes three categories of resources: a) physical, nonreconfigurable resources ( $\mathcal{PN}$ ) such as input/output (I/O) ports; b) physical, reconfigurable resources ( $\mathcal{PR}$ ) such as heaters, detectors, and regions to manipulate magnetic beads; and c) virtual, reconfigurable resources ( $\mathcal{VR}$ ) such as mixers. The set of chip resources  $\mathcal{R}$  is defined as  $\mathcal{R} = \mathcal{PN} \cup \mathcal{PR} \cup \mathcal{VR}$ . Unlike  $\mathcal{VR}$ , the resources in  $\mathcal{PR}$  and  $\mathcal{PN}$  are spatially fixed, but a resource in  $\mathcal{PR}$  can be reconfigured to leverage the electrodes located within its region for sample processing in addition to its original function. For example, a magnet resource can be used either for magnetic-bead snapping or for sample processing, but not both at the same time.

Consequently, a biochip resource  $\mathcal{R}_r \in \mathcal{R}$  is characterized by  $\mathcal{R}_r = (\gamma_r, x_r, y_r)$  where  $\gamma_r$  is the resource type, and  $x_r$  and  $y_r$  are the  $x$  and  $y$  coordinates of the resource interface, respectively. The resource interface is represented by an electrode that connects the resource to the global, unidirectional routing bus. Fig. 13 shows an example of DMF resources. A dedicated dispensing reservoir is used to store a replenishment solution to counter droplet evaporation [33]. The proposed chip layout can implement the epigenetic regulation protocol, and it facilitates the real-time coordination of multiple droplets along the global routing bus.

2) *Fluidic Operations in Multiple Pathways*: In prior work, bioassay operations and the interdependencies among them have been modeled as a directed sequencing graph  $\mathcal{G} = (V, E)$  [41]. A node  $v \in V$  signifies an operation and an





**Fig. 13. Resources of a DMF system used for implementing the epigenetic regulation protocol.**

edge  $e = (v_1, v_2) \in E$  represents precedence relation between operations  $v_1$  and  $v_2$ , respectively. This model is sufficient to handle synthesis for a single sample pathway, but inadequate for multiple sample pathways.

When multiple sample pathways are involved, modeling the synthesis problem based on real-time system theory enables us to assess system performance and robustness under various conditions and constraints, e.g., through compositionality and schedulability analyses [76]. In analogy with real-time multiprocessor scheduling, we introduce the following key terms to model the synthesis problem.

- Fluidic-task set ( $\mathcal{T}$ ): Similar to computational tasks in computer systems, a fluidic task  $\tau_b \in \mathcal{T}$  represents a bioassay in a target protocol. A fluidic subtask  $\tau_b^i$  represents a fluid-handling operation within a bioassay  $\tau_b$ .
- Fluidic-processor set ( $\mathcal{R}$ ): A biochip resource is also referred to as a fluidic processor. Hence, a DMFB is comprised of a set of heterogeneous processors corresponding to the chip resources  $\mathcal{R}$ .

In real-time systems, precedence-related subtasks can be assigned and scheduled on dedicated processors, where inter-processor communication induces delays in the release of subsequent subtasks [77]. Similarly, we build on the directed-acyclic graph (DAG) model [78] to capture the characteristics of a quantitative-analysis protocol. This model can then be used to compute task-assignment and scheduling solutions in our DMF system. The details of our task model are described below.

- The protocol consists of a set of bioassays  $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_n\}$ . Each bioassay  $\mathcal{B}_b \in \mathcal{B}$  is a fluidic task  $\tau_b$  that is characterized as a 3-tuple  $(\phi_b, G_b, D_b)$ , where  $\phi_b$  is the task release time,  $G_b$  is a DAG, and  $D_b$  is a positive integer representing the relative deadline of the task. DAG  $G_b$ , whose number of nodes will be denoted by  $z_b$ , is specified as  $G_b = (V_b, E_b, P_b, C_b)$ , where  $V_b$  is a set of vertices representing subtasks

$\{\tau_b^1, \tau_b^2, \dots, \tau_b^{z_b}\}$ ,  $E_b \in [0, 1]^{z_b \times z_b}$  is an adjacency matrix that models the directed-edge set of  $G_b$ , and  $P_b \in [0, 1]^{z_b \times z_b}$  is a matrix derived from  $E_b$  that represents precedence relationships between the vertices  $V_b$ ; formally,  $P_b(i, j) = 1$  if and only if subtask  $\tau_b^i$  has to be completed before subtask  $\tau_b^j$  starts. Finally,  $C_b \in \mathbb{N}^{z_b \times z_b \times |\mathcal{R}| \times |\mathcal{R}|}$  represents the lower bound routing-cost matrix between the vertices  $V_b$ , considering all resource combinations used for subtask executions. Specifically, if resources used to execute  $\tau_b^i$  and  $\tau_b^j$  are determined to be  $\mathcal{R}_r$  and  $\mathcal{R}_{r'}$ , respectively,  $C_b(i, j, r, r') = L$  indicates that the lower bound value for the routing distance from the interfacing electrode of  $\mathcal{R}_r$  to that of  $\mathcal{R}_{r'}$  is equal to  $L$ . Note that  $C_b(i, j, r, r')$  is a function of coordinates  $(x_r, y_r)$  and  $(x_{r'}, y_{r'})$ , and it is calculated based on the unidirectional ring bus shown in Fig. 13. Furthermore, note that  $C_b(i, j, r, r')$  does not have to be equal to  $C_b(i, j, r, r)$ .

- A fluidic subtask  $\tau_b^i \in \{\tau_b^1, \tau_b^2, \dots, \tau_b^{z_b}\}$  is characterized by  $\tau_b^i = (T_b^i, \alpha_b^i, S_b^i, F_b^i)$ , where i)  $T_b^i$  is a vector used to specify the times needed by the chip resources to execute subtask  $\tau_b^i$ ; ii)  $\alpha_b^i \in \{0, 1\}^{|\mathcal{R}|}$  is a vector that specifies the subtask assignment to biochip resources  $\mathcal{R}$  (for example,  $\alpha_b^i(r) = 1$  if subtask  $\tau_b^i$  is allocated to the resource  $\mathcal{R}_r$ ); iii)  $S_b^i$  represents the start time of the subtask; and iv)  $F_b^i$  represents the end time of the subtask.

Thus, to satisfy the requirements of the bioassay  $\mathcal{B}_b$ , the following constraints must be satisfied:

$$\forall i, j, b : \tau_b^i, \tau_b^j \in \{\tau_b^1, \dots, \tau_b^{z_b}\}; r, r' : \mathcal{R}_r, \mathcal{R}_{r'} \in \mathcal{R}.$$

$$(C1) \sum_r \alpha_b^i(r) = 1 \text{ \{allocation\};}$$

$$(C2) F_b^i \geq S_b^i + T_b^i(r) \cdot \alpha_b^i(r) \text{ \{task duration\};}$$

$$(C3) S_b^j - F_b^i \geq C_b(j, i, r, r') - K(2 - \alpha_b^i(r) - \alpha_b^j(r')) - K(1 - P_b(j, i)) \text{ \{task precedence and interprocessor communication\},}$$

where constant  $K$  is a large positive constant that linearizes the AND Boolean terms in the task-precedence satisfaction problem, described in

$$(S_b^j - F_b^i \geq C_b(j, i, r, r')), \text{ if } (\alpha_b^i(r')) \wedge (\alpha_b^j(r)) \wedge (P_b(j, i)). \quad (2)$$

- In order to counter droplet evaporation, replenishment steps are incorporated before the start of every new bioassay [33]. Note that each bioassay  $\mathcal{B}_b$  (composed of a set of subtasks  $\tau_b^i$ ) is characterized by a deadline  $D_b$  on its completion time after which a sample must be run through a just-in-time replenishment process. Note that fulfilling a bioassay deadline is a soft real-time requirement. In addition, the need to fulfil all protocol deadlines imposes temporal constraints on our design, since resorting to extra replenishment steps during protocol execution significantly

<sup>2</sup> $T_b^i(r) = \infty$  indicates that subtask  $\tau_b^i$  cannot execute on  $\mathcal{R}_r$ .

impacts completion time. In other words, if a fluidic task violates its deadline (creating nonzero tardiness), extra replenishment steps are applied to the associated sample pathway to counter droplet evaporation, leading to an increase in completion time.

To address this problem, each subtask  $\tau_b^i$  is also characterized by a Boolean variable  $\Omega_b^i \in \{0, 1\}$ , where  $\Omega_b^i = 1$  indicates that subtask  $\tau_b^i$  is planned to start execution after the deadline  $D_b$ . In addition,  $\pi_b$  represents the start time of a bioassay  $\mathcal{B}_b$  and  $Q$  models the number of time steps<sup>3</sup> needed to complete sample replenishment. The following additional constraints must be satisfied:

$$\forall i, j, b : \tau_b^i, \tau_b^j \in \{\tau_b^1, \dots, \tau_b^{z_b}\}; r, r' : \mathcal{R}_r, \mathcal{R}_{r'} \in \mathcal{R}.$$

(C4)  $S_b^i \geq \pi_b$  {bioassay start time}. The value of the variable  $\Omega_b^i$  for every subtask  $\tau_b^i$  can be specified using the difference between the subtask start time  $S_b^i$  and the absolute deadline  $(\pi_b + D_b)$  of the bioassay, as follows.

(C5)  $U \cdot \Omega_b^i \geq S_b^i - (\pi_b + D_b)$ ;  $\Omega_b^i \geq 0$ ;  $\Omega_b^i \leq 1$  {tasks beyond deadline}, where  $U$  is a large positive constant that can be used to upper bound the allowable range of tardiness.

(C6)  $F_b^i \leq T_{PF}$  {protocol finish time}. In addition, the inequality in (C3) is modified to incorporate replenishment as follows. Note that just-in-time replenishment is applied before subtask  $\tau_b^i$  only when  $\Omega_b^i = 1$  and  $\Omega_b^i = 0$ , such that  $E_b(j, i) = 1$ . In other words, the bioassay deadline  $D_b$  is violated during or immediately after the execution of  $\tau_b^i$ .

(C3')  $S_b^i - F_b^i \geq C_b(j, i, r, r') - K(2 - \alpha_b^i(r) - \alpha_b^i(r')) - K(1 - P_b(j, i)) + Q \cdot \lambda_b^{(j,i)}$ , where  $\lambda_b^{(j,i)}$  is a Boolean variable specified through the expression  $(\lambda_b^{(j,i)} = \neg \Omega_b^i \wedge \Omega_b^i \wedge E_b(j, i))$ , and it can be formulated as in (C7).

(C7)  $\lambda_b^{(j,i)} \geq \Omega_b^i - \Omega_b^i + E_b(j, i) - 1$ ;  $\lambda_b^{(j,i)} \leq \Omega_b^i$ ;  $\lambda_b^{(j,i)} \leq 1 - \Omega_b^i$ ;  $\lambda_b^{(j,i)} \leq E_b(j, i)$ ;  $\lambda_b^{(j,i)} \geq 0$  {replenishment requirement}.

- Finally, to achieve mutual exclusion in system resources, each resource  $\mathcal{R}_r \in \mathcal{R}$  is also characterized by a Boolean variable  $\omega_{(i,b)}^r(t)$ , where  $\omega_{(i,b)}^r(t) = 1$  indicates that  $\mathcal{R}_r$  is being utilized by subtask  $\tau_b^i$  at time  $t$ . Therefore, the following constraint must be satisfied.

(C8)  $\sum_{(i,b)} \omega_{(i,b)}^r(t) = 1$  {mutual exclusion}. The value of

$\omega_{(i,b)}^r(t)$  can be specified as follows.

(C9)  $\omega_{(i,b)}^r(t) \geq 1 - K(3 - \alpha_b^i(r) - \delta_b^i(t) - \eta_b^i(t))$ , where  $\delta_b^i(t)$  and  $\eta_b^i(t)$  are Boolean variables that are specified through the following formulation.

(C10)  $U \cdot \delta_b^i(t) \geq t - S_b^i$ ;  $\delta_b^i(t) \geq 0$ ;  $\delta_b^i(t) \leq 1$ .

(C11)  $U \cdot \eta_b^i(t) \geq F_b^i - t$ ;  $\eta_b^i(t) \geq 0$ ;  $\eta_b^i(t) \leq 1$ .

### C. Task Assignment and Scheduling

In this section, we present our algorithm for fluidic task assignment and scheduling.

<sup>3</sup>A time step refers to the clock period, typically in the range of 0.1–1 s [79].

- 1) **Problem Formulation: Inputs:** i) A set of bioassays  $\mathcal{B}$ , where each bioassay  $\mathcal{B}_b \in \mathcal{B}$  is characterized by a task  $\tau_b \in \mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_n\}$ , bioassay deadlines  $D = \{D_1, D_2, \dots, D_n\}$ , and adjacency matrices  $\{E_1, E_2, \dots, E_n\}$ . ii) Precedence-relationship matrices  $\{P_1, P_2, \dots, P_n\}$ . iii) A set of subtasks  $\{\tau_b^1, \tau_b^2, \dots, \tau_b^{z_b}\}$  for each task  $\tau_b \in \mathcal{T}$ . iv) The routing-cost matrix  $C_b$  for each task  $\tau_b$ . v) The processing-time vector  $T_b^i$  for each subtask  $\tau_b^i$ . vi) Biochip resources  $\mathcal{R}$ .

**Output:** i) Assignment of fluidic subtasks to resources  $\alpha_b^i(r)$ . ii) Start time  $S_b^i$  and finish time  $F_b^i$  for each subtask  $\tau_b^i$ .

**Objective:** Reduce the number of tardy tasks to avoid repetition of droplet-replenishment procedures.

2) **Heuristic Algorithm:** The scheduling problem addressed here is mapped to the problem of scheduling DAG tasks on heterogeneous multiprocessors. Recently, schedulability of DAG tasks in uniform multiprocessors has been investigated and it has been shown that this problem is NP-hard [78], [80]. Not much is known about schedulability of DAG tasks in heterogeneous multiprocessors, but the problem is likely to be computationally intractable [77]. We handle this problem as described below.

The pseudocode for the algorithm is described in Algorithm 1. The algorithm expects a user-specified input that characterizes the upper bound ( $\Upsilon$ ) on the number of iterations (Line 4) to terminate the optimization process. In addition,  $\rho$  and  $\sigma$  are integers that are used as weighting

**Algorithm 1** Task Assignment and Scheduling

---

```

1: procedure Main()
2:    $Sol \leftarrow \emptyset$ ;  $T_c \leftarrow \infty$ ;  $vl \leftarrow \infty$ ;  $util_{min} \leftarrow \infty$ ;
3:    $iter \leftarrow 0$ ;  $\alpha_{prev} \leftarrow \emptyset$ ;
4:   while  $iter \leq \Upsilon$  do
5:      $iter \leftarrow 0$ ;
6:      $\alpha \leftarrow$  PROCESSOR_ASSIGN();
7:      $T_{PF}, S_b^i, F_b^i \leftarrow$  COORDINATE();
8:      $vl \leftarrow$  CAL_VIOLATIONS();
9:      $util \leftarrow \rho \cdot T_{PF} + \sigma \cdot vl$ ;
10:    if  $util < util_{min}$  then
11:       $util_{min} \leftarrow util$ ;
12:       $Sol \leftarrow$  GET_RESULTS();
13:    end if
14:     $\alpha_{prev} \leftarrow \alpha_{prev} \cup \alpha$ ;  $iter \leftarrow iter + 1$ ;
15:  end while
16:  return  $Sol$ ;
17: end procedure
18: procedure COORDINATE()
19:    $t \leftarrow 0$ ;
20:   while true do
21:      $RQ \leftarrow$  PUSH_READY_TASK_SORT( $\tau, P, C$ );
22:     if  $t == D_b$  AND TASK_UNFIN( $\tau_b, E$ ) then
23:       REPLENISH( $\tau_b, Q$ );
24:     end if
25:      $S_b^i, F_b^i \leftarrow$  SCHEDULE_IF_POSSIBLE( $RQ$ );
26:      $t \leftarrow t + 1$ ;
27:     if all  $\mathcal{T}$  scheduled then break;
28:   end while
29:   return  $\{T_{PF}, S_b^i, F_b^i\}$ ;
30: end procedure

```

---

factors for the utility function (Line 9), which is used to determine the goodness of a solution.

At every iteration, the algorithm performs task assignment and scheduling (Lines 6–7); subtasks are randomly assigned to the fluidic processors according to the required resources (Line 6), whereas a scheduling algorithm is used for real-time scheduling (Lines 18–30). The algorithm evaluates the utility of the produced solution based on the number of bioassay-deadline violations ( $v_i$ ) and the total completion time ( $T_{PF}$ ) for the protocol (Lines 8–9). The solution with the best utility (Lines 10–13) is finally selected. We introduce two policies to guide the behavior of the scheduler (Line 21) below for updating the ready subtask queue  $RQ$ . Note that a scheduling policy must preserve precedence relationships among subtasks. In addition, a scheduling decision taken by a policy must take into consideration the droplet-routing cost (interprocessor communication cost).

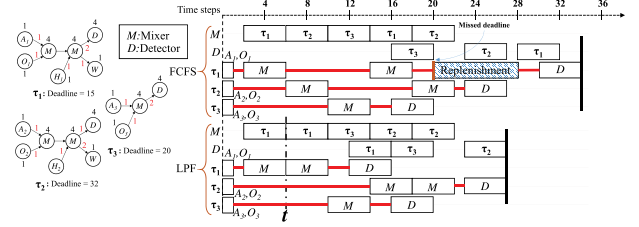
1) *First-Come-First-Served (FCFS)*: A static policy in which the ready subtasks are prioritized based on their resource-request time. Note that the resource-access time for a subtask  $\tau_b^i$  depends on the finish time of the preceding subtasks. This static approach is oblivious to bioassay deadlines.

2) *Least-Progression-First (LPF)*: A dynamic policy in which the task that belongs to the least-progressing bioassay is selected first. The least-progressing bioassay is a bioassay that is most likely to miss its deadline and its tasks urgently need to be advanced. This policy is similar to the least-laxity-first policy [81]. Quantifying the progression of bioassay execution is performed through a utility function  $f(D_b, t, s, n) = (D_b - t)(1 - (s/n))$ , where  $D_b$  is the bioassay deadline,  $t$  is the elapsed time since the bioassay has started,  $s$  is the number of time steps completed by this bioassay, and  $n$  is the summation of the number of time steps for all the bioassay operations. Note that the least-progressing bioassay has the lowest utility value.

We are given a set  $RQ$  of ready subtasks. We determine the processing time, the start time, and the finish time of every subtask in  $RQ$  and ensure that a subtask can get hold of the preassigned resource at time  $t$  (Line 25). Note that a task  $\tau_b$  that is not completed by the deadline is suspended until sample replenishment is carried out (Lines 22–24). Based on the scheduling choices, the synthesis algorithm (using the one-pass algorithm in [82]) is invoked to generate the actuation sequences (Line 12).

The scheduling scheme developed in this work is based on a timewheel that is controlled by an entity known as the coordinator and a priority queue, which is used to enforce the policy [75].

3) *Scheduling-Policy Analysis*: We analyze the scheduling policies explained above using the example in Fig. 14. We consider two cases: case A) a case where a limited-resource chip is given; case B) a case where an unlimited-resource chip is given. In case A, we consider a biochip with a single mixer



**Fig. 14. Illustration of scheduling fluidic tasks using FCFS and LPF policies with a limited-biochip setting. Red lines indicate interprocessor communication costs (in time steps).**

$M$ , a single optical detector  $D$ , and a single waste reservoir  $W$  that is used to discard droplets. In both case A and case B, we consider a dedicated reservoir for each dispensing operation; i.e., unique resources  $A_b$ ,  $O_b$ , and  $H_b$  for each task  $\tau_b$ . The task set consists of three tasks  $\tau_1$ ,  $\tau_2$ , and  $\tau_3$ , which have release times  $\phi_1 = \phi_2 = \phi_3 = 0$ . The DAG representation for these tasks is shown in Fig. 14. The letters inside the nodes indicate the assigned resources. Also, the numbers above the nodes (shown in black) represent the worst case processing time resulting from task assignment, and the numbers shown in red represent the droplet-routing cost. We consider that the replenishment steps are carried out using dedicated resources in both case A and case B, and the time needed to complete droplet replenishment is eight time steps.

In Fig. 14, we demonstrate the timeline of the scheduling output for FCFS and LPF when case A is considered. The results show that LPF outperforms FCFS for a limited-biochip setting. The reason is that LPF dynamically adapts the priorities of tasks based on their progression. For example, at time  $t$  (shown in Fig. 14), all tasks are competing for  $M$ . The utility values  $f$  of  $\tau_1$ ,  $\tau_2$ , and  $\tau_3$  at this time are 5.625, 22.75, and 11.2, respectively. As a result,  $\tau_1$ , with the least utility, is selected to process upon  $M$  starting at  $t$ . The completion times for FCFS and LPF are 34 and 27, respectively.

Nevertheless, we expect that both approaches converge to an equivalent lower bound completion time when we increase the number of on-chip resources. Given the same set of tasks and considering case B, the number of biochip resources is  $\sum_{b=1}^n z_b = 7 + 7 + 4 = 18$ ; i.e., there is a dedicated resource for each subtask  $\tau_b^i$ . In this case, it is easy to demonstrate that task scheduling depends only on the timing characteristics of a given task set, and that the computed completion time represents the lower bound, based on a specific task assignment. We introduce the following definition that aids in our explanation [78].

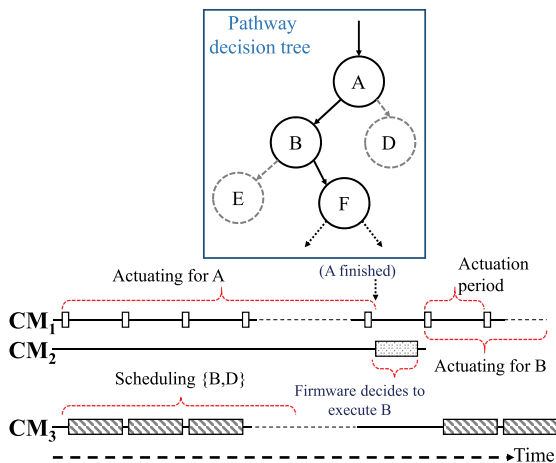
In case B, the completion times obtained by FCFS and LPF are equal and they can be defined using  $T_{PF}^* = \max len(G_b)$ . According to the task set given<sup>4</sup> in Fig. 14,  $T_{PF}^* = \max_{b \in G_b} (17, 17, 12) = 17$ , which is the lower bound completion time for FCFS and LPF when the number of resources

<sup>4</sup>Consider a unit routing cost between the two  $M$  operations in  $\tau_1$  and  $\tau_2$ .

are increased; this finding is corroborated using simulations in Section V-D1.

The worst case time complexity of the above algorithm is  $O(\Upsilon \cdot n \cdot |R|)$  when FCFS is used and  $O(\Upsilon \cdot n^2 \cdot |R|)$  when LPF is used. The algorithm is invoked whenever there is a decision that has been taken, necessitating a change in the task assignments and schedules for the sample pathways. Note that the timing overhead of this algorithm is based on the value of the optimization parameter  $\Upsilon$ , the number of on-chip resources, and the total number of protocol time steps. Since both the number of resources and time steps are specified in advance for a given platform, a biochip user can modify the response characteristics of a microfluidic system only by changing the value of  $\Upsilon$ . In our microcontroller demonstration, described in Section V-D3, we set  $\Upsilon$  to 1 to ensure faster responses to protocol decisions.

To make use of the hierarchical system structure in Fig. 12, the execution of the algorithm needs to be interleaved with electrode actuation and firmware computation. An approach for employing this scheme is to run the task-assignment and scheduling algorithm in a lookahead manner. The progression of a sample pathway through a sequence of bioassays, based on the flow decisions, is modeled as a decision tree. While  $CM_1$  executes a bioassay at the  $q$ th level,  $CM_3$  concurrently carries out task assignment and scheduling considering all choices at the  $(q+1)$ th level. Subsequently, the appropriate assignments and schedules are selected based on the detection results obtained from  $CM_2$ . Note that  $CM_3$  must complete computation of the  $(q+1)$ th level before  $CM_1$  finishes the execution at the  $q$ th level. Fig. 15 depicts the timeline of a lookahead-based execution for a pathway. This execution mechanism can significantly reduce the impact of the timing overhead; the algorithm is executed concurrently with biochip actuation, which is usually configured to function at a rate of 1–100 Hz.



**Fig. 15. Lookahead-based execution for a sample pathway. The progression of the sample is modeled as a decision tree.**

## D. Simulation Results and Experimental Demonstration

We implemented the proposed heuristic algorithm using C++. The set of bioassays of the quantitative gene-regulation protocol (described in Section V-A) were used as a benchmark. A comprehensive analysis of the algorithm is performed based on three groups of evaluations: 1) evaluation of scheduling policies using simulation-based analysis; 2) comparison with previous resource-allocation techniques; and 3) real-time experimental demonstration using an embedded microcontroller.

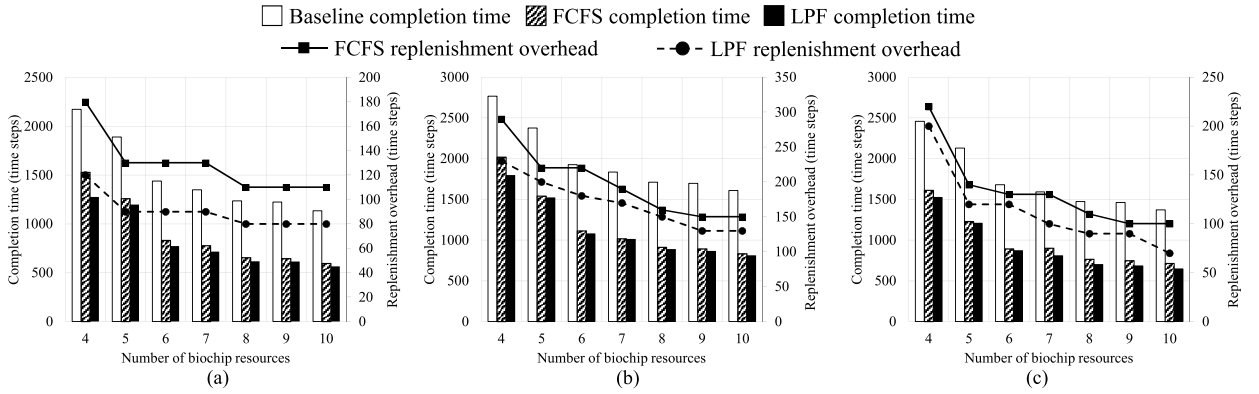
1) *Evaluation of Scheduling Policies:* Since most of the previous design-automation methods have not considered multiple sample pathways, we derive a baseline in which the protocol is executed for each sample separately. Therefore, we evaluate the performance for three task-scheduling schemes: a) the baseline; b) FCFS; and c) LPF. The metrics of comparison include: the total completion time for the protocol (including replenishment time) and the time overhead incurred by replenishment procedures due to deadline violation; all measured in time steps. The results were obtained for various chip sizes, which are represented in terms of the number of biochip resources (e.g., heaters, mixers, and detectors); see Fig. 13. The deadlines for bioassays were obtained using offline simulation—each bioassay was simulated for various chip sizes and then the longest completion time was considered as a deadline.

We consider three samples being concurrently subjected to fluidic operations. The samples are S1 (GFP gene-targeted sample), S2 (YFP gene-targeted sample), and S3 (actin gene-targeted sample). We consider three different cases in terms of the sample pathways: Case I (short homogeneous pathways): a case where all three samples follow the same shortest pathway (12 bioassays in each pathway); Case II (long homogeneous pathways): a case where all three samples follow the same long pathway (16 bioassays in each pathway); and Case III (heterogeneous pathways): a case where these samples are different (the three pathways comprise 12, 14, and 16 bioassays, respectively).

Fig. 16 compares the three scheduling schemes in terms of completion times for the three cases. Although the baseline scheme does not provide resource sharing (hence no deadline violation occurs), it leads to the highest completion times for all chip sizes. We also observe that LPF provides lower completion time compared to FCFS for tight resource constraints. This result corroborates our analysis in Section V-C3, in which we demonstrated that LPF is a deadline-driven approach and it achieves less tardiness, compared to FCFS.

We also estimate the overhead incurred due to replenishment when FCFS and LPF are used. Without loss of generality, we assume that a replenishment procedure takes ten time steps in the worst case. It is obvious that LPF incurs less replenishment overhead for all chip sizes, as shown in



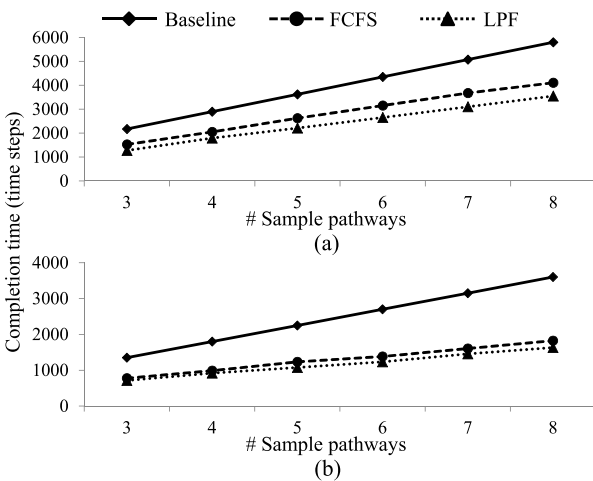


**Fig. 16. Completion times and replenishment overhead for three task-scheduling schemes—baseline, FCFS, and LPF—running (a) short homogeneous pathways, (b) long homogeneous pathways, and (c) heterogeneous pathways.**

Fig. 16. As a result, compared to FCFS, the priority scheme of the LPF scheduler is more effective in countering droplet evaporation while the protocol completion time is not significantly increased. Also, it is noteworthy to mention that the difference in protocol completion time between LPF and FCFS gradually decreases when the number of biochip resources is increased; see Fig. 16. This finding also adheres to our analytical study performed in Section V-C3.

Next, we explore scalability of the three scheduling schemes. We analyze the completion time as the number of homogeneous pathways is varied; see Fig. 17. We observe that, as expected, the baseline scheme does not scale with the number of samples, since its completion time increases considerably when the number of samples is increased.

2) *Comparison With Resource Allocation in Section VI:* We next compare the completion time and resource utilization of our real-time scheduling approach (using LPF) with



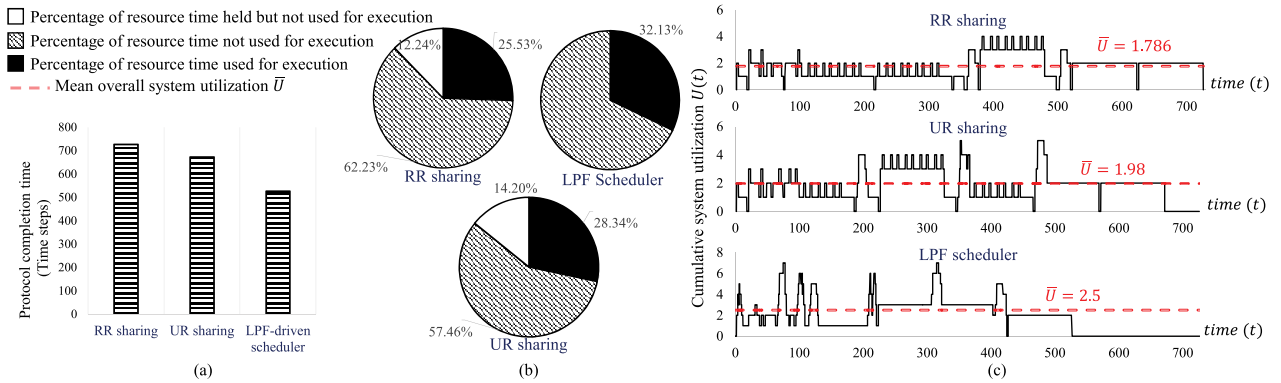
**Fig. 17. Scalability of the scheduling schemes with a varying number of homogeneous pathways: (a) using a biochip with four resources; and (b) using a biochip with seven resources.**

resource-allocation techniques proposed in Section IV. This work introduced two schemes to derive upper and lower bounds on protocol completion time considering resource sharing. These schemes are: a) restricted resource sharing (RR): a scheme that fully restricts the reconfigurability of shared resources among bioassays; and b) unrestricted resource sharing (UR): a scheme that does not consider any restrictions on resource sharing.

For comparison, we use a simulation environment similar to that in Section IV, in which a gene-expression analysis protocol is used. The minimum resource requirements as well as fluid-handling operations are reused from Section IV. We utilize three short homogeneous sample pathways, but the conclusion of this simulation holds also for other cases. The biochip architecture consists of seven resources: two mixers, two optical detectors, one magnet, one heater, and one charge-coupled device (CCD) camera. The utilization profile of a resource  $\mathcal{R}_r$  over the course of the protocol execution is denoted by  $u_r(t)$ , where  $u_r(t) = 1$  indicates that resource  $\mathcal{R}_r$  is used for execution during the time step  $[t, t + 1]$ , whereas  $u_r(t) = 0$  indicates that the resource is idle during the same time step. The summation of the utilization profiles of all chip resources is referred to as cumulative system utilization  $U(t)$ ; thus  $U(t) = \sum_{r=1}^7 u_r(t)$ . This function gives an indication of how efficiently chip resources are used by a resource-allocation scheme over time. We use this function to derive the mean overall system utilization  $\bar{U}$  to compare system utilization of resource-allocation schemes using numerical values. The parameter  $\bar{U}$  is calculated as follows:

$$\bar{U} = \frac{\int_0^{T_{PF}} U(t) \cdot dt}{T_{PF}} \quad (3)$$

where  $T_{PF}$  is the protocol completion time. An upper bound for  $\bar{U}$  is equal to the number of chip resources; i.e.,  $\bar{U} \in [0, 7]$ . Note that higher  $\bar{U}$  signifies better resource utilization. Although our real-time scheduler takes into



**Fig. 18. Comparison between three resource-management schemes: restricted resource sharing (RR), unrestricted resource sharing, and LPF-based real-time scheduler (proposed). Comparison is based on: (a) protocol completion time, (b) percentage of time effectively used for execution, and (c)  $U(t)$  and  $\bar{U}$ .**

consideration droplet-routing overhead, we ignore this cost in our comparison.

Fig. 18(a) compares the three resource-management schemes based on the completion time for gene-expression analysis. It is obvious that the proposed real-time scheduler, i.e., the LPF scheduler, achieves the shortest completion time, compared to the resource-allocation schemes from Section IV. As expected, RR sharing leads to the worst case completion time due to the restrictions imposed on resource sharing.

Next, we analyze the resource utilization that results from the resource-management schemes, based on two metrics: percentage of resource time used effectively by protocol execution; and  $\bar{U}$ . The results for the former metric is illustrated in Fig. 18(b). Using the LPF scheduler, 32.13% of the overall resource time is effectively used to execute protocol fluid-handling operations, whereas only 25.53% and 28.34% of the overall resource time are effectively used by RR sharing and UR sharing, respectively. Note that resource allocation in Section IV is carried out at the bioassay level; hence, a set of resources might be reserved to execute a bioassay, but not used until the associated fluid-handling operations are invoked. As a result, both RR sharing and UR sharing incur a time cost due to reserving, but not using, chip resources; see Fig. 18(b).

Finally, results involving the parameter  $\bar{U}$  for resource utilization are shown in Fig. 18(c). The proposed LPF scheduler achieves the best cumulative system utilization ( $\bar{U} = 2.5$ ), compared to RR sharing ( $\bar{U} = 1.786$ ) and UR sharing ( $\bar{U} = 1.98$ ), respectively. These results indicate that the proposed real-time scheduler is more cost effective and it is applicable for tighter resource-budget cases.

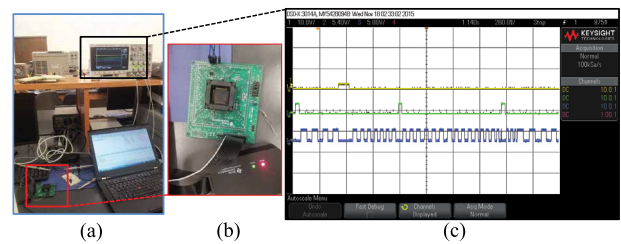
3) *Experimental Demonstration:* We next demonstrate the application of hierarchical scheduling using a commercial off-the-shelf micro-controller [TI 16-bit MSP430 100-pin target board; see Fig. 19(b)]. An oscilloscope is used to probe signals generated from the components  $CM_1$ ,  $CM_2$ , and  $CM_3$ ; the experimental setup is shown in Fig. 19(a). The

generated waveform is shown in Fig. 19(c), where the blue signal indicates the synthesized time steps based on the task scheduler ( $CM_3$ ), the green signal represents the actuation pulses ( $CM_1$ ), and the yellow pulse reflects firmware computation ( $CM_2$ ) based on a detection operation.

Our experimental target was to execute gene-expression analysis using two sample pathways. For testing and verification purposes, we generate a hypothetical, but feasible, stream of data to mimic the data transfer between the hardware and the control software. These data were obtained by simulating the gene-expression analysis protocol with two sample pathways.

A timer-interrupt module was utilized to trigger periodic actuation tasks while the task scheduler was running. This mechanism provides flexible tuning for the actuation clock depending on the developed application [79]. The firmware tasks were also realized through a set of interrupt service routines (ISRs) and invoked after each (simulated) detection operation. The selection of an ISR depends on the type of detection method assumed, e.g., CCD-camera monitoring of cell culture or fluorescence detection of amplified nucleic acid.

The outcome of the experiment matches the lookahead timing model described in Section V-C2. This experiment lays the foundations for developing and testing the key hardware/software codesign components for real-time



**Fig. 19. Experimental demonstration of hierarchical scheduling. (a) Experimental setup. (b) The 16-bit MSP430 100-pin target board. (c) Probed signals from  $CM_1$  (green),  $CM_2$  (yellow), and  $CM_3$  (blue).**

DMFBs that can be used in realistic microbiology protocols. Hence, the work proposed in this section adds another key component, which is the real-time component, to the synthesis infrastructure established in Section IV. As a result, similar to multicore system-on-chips, mapping the resource allocation of multiple samples (Section IV) to the real-time context offers modular and channel-based integration between a DMFB and other onboard analysis modules such as DNA sequencers.

## VI. SYNTHESIS FOR PROTOCOLS WITH INDEXED SAMPLES: REALIZATION OF TYPE-DRIVEN SINGLE-CELL ANALYSIS

Single-cell analysis using affordable microfluidic technologies has now become a reality [65], [83]. Thousands of heterogeneous cells can be explored in a high-throughput manner to investigate the link between gene expression and cell types, thereby providing insights into diseases such as cancer [84]. Microfluidic techniques have recently been developed to conduct each step of the following single-cell experimental flow.

- 1) Cell encapsulation and differentiation: Heterogeneous cells are isolated, encapsulated inside droplets, and differentiated according to their identity (type); e.g., their shape, size, cell-cycle stage, or lineage.
- 2) Droplet indexing (barcoding): Each droplet is manipulated through a sequence of biochemical procedures such as cell lysis and mRNA analysis. At the end of these steps, the *in situ* type of the encapsulated cell may no longer be available for downstream analysis [85]. Therefore, indexing of droplets using barcodes is needed to keep track of their identity.
- 3) Type-driven cell analysis: Single-cell bioassays such as chromatin immunoprecipitation (ChIP) are carried out using microfluidics, where the selection of a bioassay relies on the cell type that is identified in Step 1 [65]. To draw meaningful conclusions, the experimental outcomes are associated with droplet barcodes injected in Step 2 [62].

Although the synthesis methods in Sections IV and V can handle multiple independent sample pathways, there are two barriers that need to be addressed in order to adopt these methods for practical single-cell studies.

- Heterogeneity of single-cell methods: Not all the above biochemical steps can be efficiently miniaturized using a single microfluidics technology. Valve-based techniques are used to rapidly separate and isolate biomolecules with high resolution, making them suitable for cell encapsulation (Step 1) [65]. On the other hand, digital-microfluidic biochips (DMFBs) enable real-time decision making for sample processing and genomic-analysis protocols, such as quantitative polymerase chain reaction (qPCR) [34] (Step 3).

However, DMFBs are not as effective for interfacing to the external world [86]. Hence, there is a need for a hybrid microfluidic system that combines the advantages of the two domains, and a synthesis method that controls single-cell experiments in a dual-domain microfluidic setting.

- Scalable droplet indexing: A single-cell analysis flow may involve hundreds of cell types, each of which requires a distinct barcode for downstream analysis using digital microfluidics. Therefore, droplet indexing on a DMFB requires either the use of prestored droplets that host individual barcoding hydrogels [62] (or dyes)<sup>5</sup>—not feasible when a large number of cells are being investigated—or a specific input reservoir for each cell type. The latter solution increases the fabrication cost dramatically. Furthermore, since reservoir control is not readily automated [87], it is unrealistic to assume that each dispensed droplet contains only one barcoding particle. Therefore, there is a pressing need for a low-cost mechanism for droplet barcoding.

In Sections IV and V, we introduced a synthesis framework for DMFBs to support multiple sample pathways, but we considered that sample differentiation and indexing (Step #2 in Fig. 4) have been carried out in advance before protocol execution. Therefore, the earlier frameworks overlooked the above challenges.

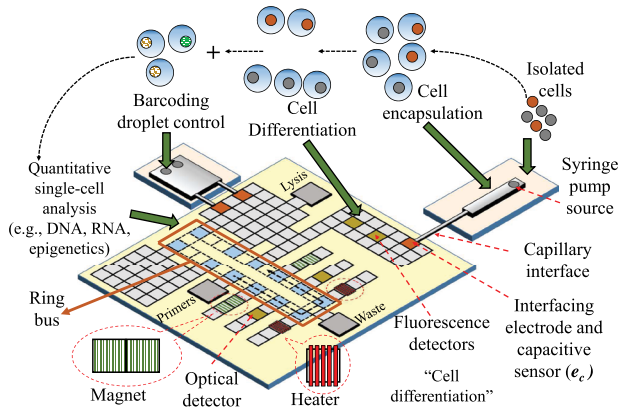
In this section, we address the above challenges by introducing a hybrid and cyber-physical microfluidic platform for integrated single-cell analysis. We present a synthesis method, referred to as Co-Synthesis (CoSyn), that allows synthesis for concurrent sample pathways [83]. CoSyn enables coordinated control of the hybrid microfluidic components.

### A. Hybrid Platform and Single-Cell Analysis

Similar to the microfluidic implementation of gene-expression analysis (Section IV), single-cell analysis relies on the concurrent manipulation of sample droplets, where each sample cell is run through the protocol flow discussed earlier. An efficient on-chip implementation of the single-cell analysis protocol is accomplished using a hybrid platform and cyber-physical adaptation. Fig. 20 shows the platform components matched with different protocol stages. The two domains are connected through a capillary interface; this technique has been successfully adopted in practice [87].

1) *Cell Encapsulation and Flow Control*: As shown in Fig. 20, on-chip operation starts with the encapsulation of single cells in droplets, which is efficiently accomplished using flow-based microfluidics [87]. The droplet generator uses a syringe pump such that the flow rate of pressure-driven droplets can be automatically controlled via feedback. A capacitive sensor is placed at the interfacing electrode

<sup>5</sup>Henceforth, we refer to dyes as barcoding droplets.



**Fig. 20. The hybrid platform for single-cell analysis.**

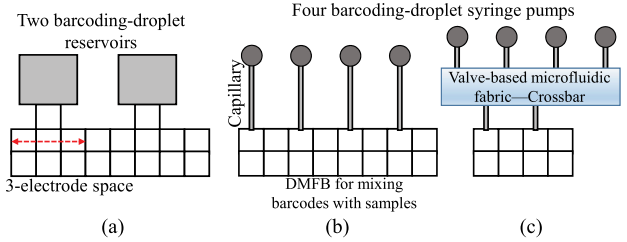
( $e_c$  in Fig. 20) on the digital side to sense a droplet [88]. When the digital array is unable to accommodate additional droplets, it stops the flow by switching off the pump.

Note that an actuator is used in the flow-based component, whereas a sensor is placed on the digital side. To synchronize the two domains, the flow-control procedure (capacitive sensing and pump control) is invoked at the same frequency as droplet actuation in the digital domain (1–10 Hz).

2) *Cell Differentiation*: Automated cell-type identification can be achieved by analyzing signaling events in single cells *in situ*. Similar to the miniaturization of gene-expression analysis, a green fluorescent protein (GFP) reporter is used for cell differentiation. In each cell, the fluorescence intensity from the GFP (detected in real time using an on-chip fluorescence detector or imaging apparatus) is used to account for differences in expression level among cells; this is equivalent to classifying cells into functional clusters that represent cell types.

Although a valve-based biochip can also be used for cell differentiation, we consider a DMFB for this purpose in CoSyn due to its demonstrated ability to carry out high-throughput fluorescence detection and distinguish between hundreds of cell types; this feature is not supported by valve-based mechanisms.

3) *Droplet Barcoding*: Droplet barcoding is essential to maintain cell identity during downstream analysis. Since thousands of cells (and hundreds of cell types) can be involved in an analysis protocol, a barcoding droplet must be dispensed on demand, and mixed with a sample droplet and other reagents according to the cell type [62]. If we consider a population with  $n$  cell types, droplet barcoding on a digital-microfluidic array requires  $n$  reservoirs, each of which typically covers a three-electrode space. An additional electrode is needed for separation; see Fig. 21(a). In this case, to accommodate  $n$  reservoirs, a lower bound on the array perimeter is  $4n + k$  electrodes, where  $k$  is a constant that represents the number of electrodes covered by other reservoirs. This approach is therefore impractical



**Fig. 21. Droplet barcoding using: (a) a DMFB; (b) a valve-based biochip with one-to-one mapping between syringe pumps and DMFB ports; and (c) a valve-based biochip with many-to-many mapping.**

because of the significant increase in chip size with the number of target cell types. It also requires the dispensing of a single hydrogel particle per droplet; this feature has not been implemented yet using reservoir control.

To overcome the above limitations, a valve-based biochip is connected to the DMFB to exploit its pressure-driven ports that have smaller footprints than reservoirs; see Fig. 21(b).

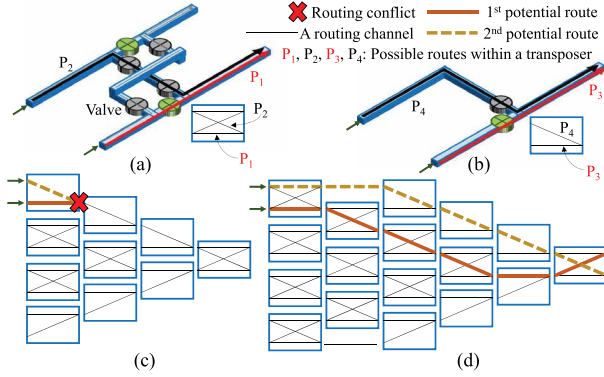
This biochip is used to generate barcoding droplets via a syringe pump; the droplets are routed to appropriate locations on the digital-microfluidic array through a capillary interface [62]. With this hybrid configuration, the lower bound on the digital-array perimeter is reduced to  $2n + k$  electrodes for  $n$  cell types. A one-electrode gap is needed to prevent accidental mixing of droplets. This hybrid configuration, however, overprovisions the number of concurrently utilized ports; it is unlikely that all cell types will simultaneously request barcoding.

As a cost-effective solution, we utilize a reconfigurable valve-based fabric [9]; see Fig. 21(c). This routing fabric acts as a crossbar since it allows routing of barcoding droplets from any of the  $n$  input ports to any of the  $m$  output ports, where  $n \gg m$ . The  $m$ -output valve-based fabric is then stitched to the DMFB; hence, the lower bound on the perimeter is decreased to  $2m + k$ . However, in this situation,  $p$  valves ( $p > 0$ ) are necessary for routing barcoding droplets across the fabric. By unlocking this capability of valve-based crossbars, we shift the scaling complexity from the digital domain to the flow-based domain, which is known to have a cost-effective fabrication process and efficient peripheral components.

We utilize the “transposer” primitive introduced in [9]. As shown in Fig. 22(a) and (b), a valve-based transposer appears in two forms: 1) a two-input, two-output transposer, which comprises six valves, controlled via two pneumatic inputs (full transposer); and 2) a two-input, one-output transposer, which consists of two valves controlled via two pneumatic inputs (half transposer). Note that only a full transposer allows simultaneous dispensing of two barcoding droplets, wherein the droplets can be driven “straight” or “crossed.”

The use of transposers to construct an  $n$ -to- $m$  valve-based crossbar leads to various design problems that must be tackled; see Table 4. An architectural design challenge





**Fig. 22. A valve-based routing fabric for droplet barcoding: (a) a two-input full transposer; (b) a two-input half transposer; (c) a four-level, 8-to-2 routing fabric; and (d) a six-level, 8-to-2 fabric.**

arises because various configurations of transposers can be exploited to achieve the required number of input and output ports. For example, an 8-to-2 crossbar can be constructed using four “vertical” levels, as shown in Fig. 22(c), or using six levels, as shown in Fig. 22(d). A six-level crossbar, while incurring higher cost, provides a higher degree of reconfigurability and flexibility in routing. Due to lack of space, we skip the architectural design challenge; we focus instead on the modeling and synthesis problems listed in Table 4.

4) *Type-Driven Single-Cell Protocol*: After a droplet is bar-coded, a single-cell analysis protocol is applied to the constituent cell in the DMFB, where protocol specifications are determined based on the cell type. For example, an investigator might be interested in identifying the gene expression of a specific genomic loci for a certain cell type A. Another cell type B might show unexpected heterochromatic state at a certain loci, and the investigator might be interested in identifying the protein interactions (i.e., causative proteins) or chromatin modifications causing this behavior. For type A, it is sufficient to perform gene-expression analysis using qPCR [34], [67], whereas ChIP protocol followed by qPCR must be used for type B to reveal the DNA strains contributing in the activity of the causative proteins [89].

## B. Mapping to Algorithmic Models

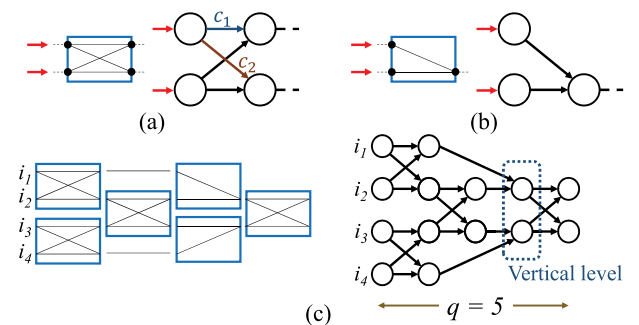
We provide a mapping of the hybrid platform and the single-cell analysis protocol to algorithmic models.

**Table 4** Design Problems for Assembling and Integrating an  $n$ -to- $m$  Valve-Based Crossbar

Problem	Objective
Architecture	Analyze the tradeoff between routing flexibility and operation timing and cost to obtain the best transposer configuration.
Modeling	Map the architecture into algorithmic semantics that support real-time routing.
Synthesis	Solve the valve-based routing problem considering real-time reconfiguration and in coordination with resource allocation in DMFBs.

1) *Modeling of a Valve-Based Crossbar*: Recall that an  $n$ -to- $m$  valve-based crossbar can be constructed using different combinations of transposers. We represent the set of transposers and their interconnections as a directed acyclic graph (DAG)  $T = (\mathcal{X}, \mathcal{Z})$ , where a vertex  $x_i \in \mathcal{X}$  is a transposer node, and an edge  $z_i \in \mathcal{Z}$  represents a connection between two transposers. Within a transposer, the point at which a droplet can be routed either straight or crossed is defined as a decision point. We map an  $n$ -to- $m$  valve-based crossbar (with a transposer network  $T$ ) into a DAG  $\mathcal{F}_{n \times m} = (\mathcal{D}_{n \times m}, \mathcal{S}_{n \times m})$ , where a vertex  $d_i \in \mathcal{D}_{n \times m}$  is a flow-decision node, and an edge  $s_i \in \mathcal{S}_{n \times m}$  represents a channel that connects two decision nodes. To simplify the discussion, we do not include  $T$  in the notation for the crossbar DAG. We can view a full (half) transposer as a 2-to-2 (2-to-1) valve-based crossbar; thus, we represent fluid-flow control in a full (half) transposer as a DAG  $\mathcal{F}_{2 \times 2}$  ( $\mathcal{F}_{2 \times 1}$ ); see Fig. 23(a)–(b). The cost  $c_i$  of  $s_i$  represents the time needed to transport fluid between the two connected nodes, measured in flow time steps ( $T_f$ ). We assume that the routing time of a droplet on a straight channel between two decision nodes is a unit of  $T_f$ . For example, as shown in Fig. 23(a),  $c_1$  is equal to  $T_f$ , whereas  $c_2$  is equal to  $2T_f$ , since even though a diagonal is shown in Fig. 22 as a fluidic path, routing of such paths in a transposer is implemented only along the  $x$ - and  $y$ -directions and the distances along these dimensions are equal [9]. Fig. 23(c) depicts the graph  $\mathcal{F}_{4 \times 2}$  for a 4-to-2 crossbar with four levels of transposers and five levels of nodes (denoted henceforth by  $q$ ;  $q = 5$  in this case).

2) *Modeling of a Digital-Microfluidic Biochip*: While DMFBs are highly reconfigurable and can support a diverse set of transport paths, we reduce the burden of managing droplet transport in real time by considering a unidirectional ring-based architecture, as shown in Fig. 20. Connected to this ring are on-chip resources. Since there is always a route between any pair of on-chip resources, a ring-based DMFB is modeled as a strongly connected DAG  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where a vertex  $v_i \in \mathcal{V}$  represents the fluid-handling operation offered by an on-chip resource, and a directed edge  $e_i \in \mathcal{E}$  represents a path (over the ring) that connects two resources. The cost  $ce_i$  of  $e_i$  indicates



**Fig. 23. Mapping a valve-based crossbar to a graph model: (a) a full transposer; (b) a half transposer; and (c) a 4-to-2 crossbar.**

the number of digital time steps ( $T_d$ ) needed to transport a droplet. We assume that the durations corresponding to  $T_f$  and  $T_d$  are equal, which can be achieved in practice by tuning the actuation frequency (Hz) and the flow rate (mL/min).

3) *Protocol Model and Cell State Machine*: To solve the synthesis problem for single-cell analysis, we take into account the complexity imposed by the barcoding mechanism. Similar to the design methodology in [67], we represent the protocol as a control flow graph (CFG)  $\mathcal{A} = (\mathcal{H}, \mathcal{L})$ , in which every node  $h_i \in \mathcal{H}$  (referred to as a supernode) models a bioassay such as qPCR; see Fig. 24. A directed edge  $l_i \in \mathcal{L}$  linking two supernodes  $\{h_j, h_k\}$  indicates that a potential decision can be made at runtime to direct the protocol flow to execute the bioassay  $h_k$  after  $h_j$ . A supernode  $h_i$ , in turn, encapsulates the sequencing graph that describes the fluid-handling operations of a bioassay and the interdependencies among them. Since there is inherent uncertainty about the type of barcoding droplets for a sample cell at design time, we extend the basic CFG model by incorporating an internal supernode (barcode propagation) that describes all possible dispensing options of barcoding droplets. Note that this model is agnostic about the type of the microfluidic technology used for implementing the protocol. Yet, the synthesis of each supernode is accomplished in a technology-aware manner using CoSyn.

In addition to the CFG model, a state machine is utilized to model the progression of each cell along the single-cell pipeline. Typically, the hybrid platform can iteratively process thousands of cells; such cells might be scattered across the platform domains at any given point in time. Therefore, this state machine (Fig. 24) is necessary to keep track of the cells that are being processed simultaneously.

### C. Co-Synthesis Methodology

This section formulates the dual-domain resource-allocation problem and describes the proposed solution (CoSyn).

1) *Problem Formulation*: Our goal is to design a fully connected fabric such that a droplet can be forwarded from any

of the  $n$  inputs to any of the  $m$  output ports. We present a sufficient criterion for achieving a fully connected fabric. The proof can be found in [90].

**Theorem 1:** An  $n$ -to- $m$ ,  $q$ -level valve-based crossbar is a fully connected fabric if  $n$  and  $m$  are even integers, and  $q \geq (m + r)/2$ .

Using this theorem, we can automatically generate the graph model  $\mathcal{F}_{n \times m}$ , thereby guaranteeing that any barcoding input can reach all  $m$  outputs. The algorithm is described in [90]. Our optimization problem is as follows.

**Inputs:** 1) The protocol CFG  $\mathcal{A}$ . 2) A matrix  $C$ . Each vector  $C_i \in C$  corresponds to a cell, and consists of integers that encode cell state machine, cell type, and the assigned bioassays in  $\mathcal{A}$ . 3) The configuration of the valve-based system. This information includes the graph  $\mathcal{F}_{n \times m}$ , the number of inputs  $n$ , and the number of outputs  $m$ . 4) The types of resources corresponding to the DMFB, their operation time, and the routing distance between each pair of resources.

**Output:** Allocation of chip modules to the individual cells and total completion time  $T_{\text{comp}}$ .

**Objective:** Minimize  $T_{\text{comp}}$  to provide high throughput.

2) *Solution Methods*: An  $n$ -to- $m$  valve-based crossbar allows only  $m$  barcoding droplets to be delivered simultaneously to the DMFB. We increase throughput by allowing pipelined routing of droplets. With pipelining, the routing algorithm allows a droplet to be routed even though a complete path to an output is unavailable. In this case, a droplet is immobilized at the farthest intermediate decision node that is not reserved by other droplets (a pipeline stage), then allowed to move forward when a path is freed. Fig. 25 illustrates pipelined and nonpipelined routing.

To solve the routing problem, we utilize a graph-theoretic algorithm to find vertex-disjoint shortest paths [91] (refer to [83] for algorithm pseudocode). By computing disjoint paths, we ensure that different barcoding droplets do not interfere with each other during routing. The routing algorithm is invoked whenever a cell transitions from the identification state to the barcoding

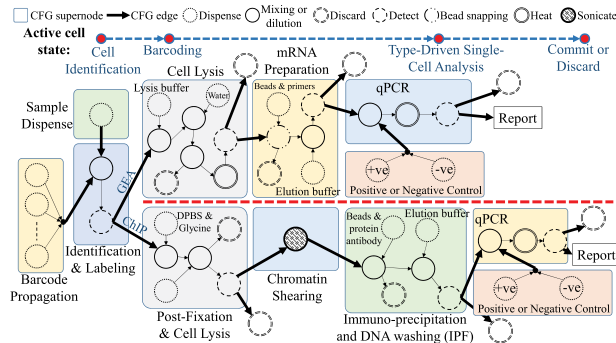


Fig. 24. CFG of type-driven analysis for a single-cell pathway.

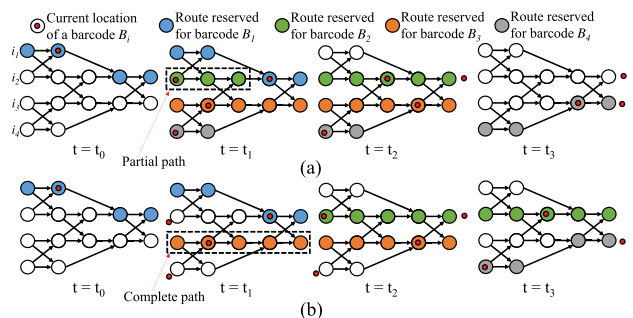


Fig. 25. Valve-based routing of four barcoding droplets (4-to-2 biochip): (a) with pipelining; (b) without pipelining.

state. If all the  $m$  outputs of the chip are currently reserved, the algorithm generates a partially disjoint shortest length path from the input source to the farthest node (Fig. 25). This is equivalent to routing the associated barcoding droplet up to an intermediate point, and holding the droplet until another disjoint path (partial or complete) can be computed to advance the droplet. The channels currently reserved for routing a barcoding droplet cannot be accessed by any other droplet until the droplet being held moves out of the valve-based crossbar.

Since the vertices of  $\mathcal{F}_{n \times m}$  are generated in topological order, the shortest paths can be quickly computed. The worst case complexity of this algorithm is  $O(|\mathcal{D}_{n \times m}| + |\mathcal{S}_{n \times m}|)$ .

We use a greedy method to solve the resource-allocation problem in the DMFB; the pseudocode can be found in [83]. We denote a DMFB resource by  $r \in R^d$ , where  $R^d$  encapsulates all DMFB resources. Thus, the cost of allocating resource  $r$  to execute a fluidic operation of type  $y \in \mathcal{Y}$  (the set  $\mathcal{Y}$  incorporates all operation types) is  $\rho(\hat{r}, r, y) = \gamma(r, y) + \varepsilon(\hat{r}, r)$ , where  $\gamma(r, y)$  is the operation time on  $r$  and  $\varepsilon(\hat{r}, r)$  is the routing distance from  $\hat{r}$  (the currently occupied resource) to  $r$ . The worst case computational complexity of this algorithm is  $O(|\mathcal{Y}|)$ .

CoSyn uses a time wheel to simulate the real-time interactions between the individual cells and the components of the hybrid system. The stages of the pipeline match the states of the cell state machine (Fig. 24). The time wheel interacts with both microfluidic domains through APIs. Whenever the time wheel locates an available fluorescence detector at the DMFB, it allocates a cell to it in order to perform type identification. Next, when the cell type is identified and there are available valve-based routes (partial or complete) to route the associated barcoding droplet, the time wheel starts the pipelined routing process through iterations until the droplet reaches the electrode interface at the digital-microfluidic side and mixed with the cell. When a DMFB resource is available to further process the cell, the previously reserved valve-based channels are released.

Real-time resource allocation for the DMFB is managed by the time wheel, which in turn, commits a cell pathway whenever its particular single-cell bioassays have executed. Based on an intermediate decision point, the cell might also be discarded during analysis.

#### D. Simulation Results

We implemented CoSyn using C++. A set of bioassays constituting the single-cell analysis protocol (Section VI-A) was used as a benchmark. Cell types were assigned to the cells using a uniform distribution function.

Since this is the first work on synthesis for hybrid microfluidic platforms, we have developed two baseline frameworks: 1) architectural baseline (ArcSyn), wherein the barcoding fabric is valveless and it utilizes a one-to-one

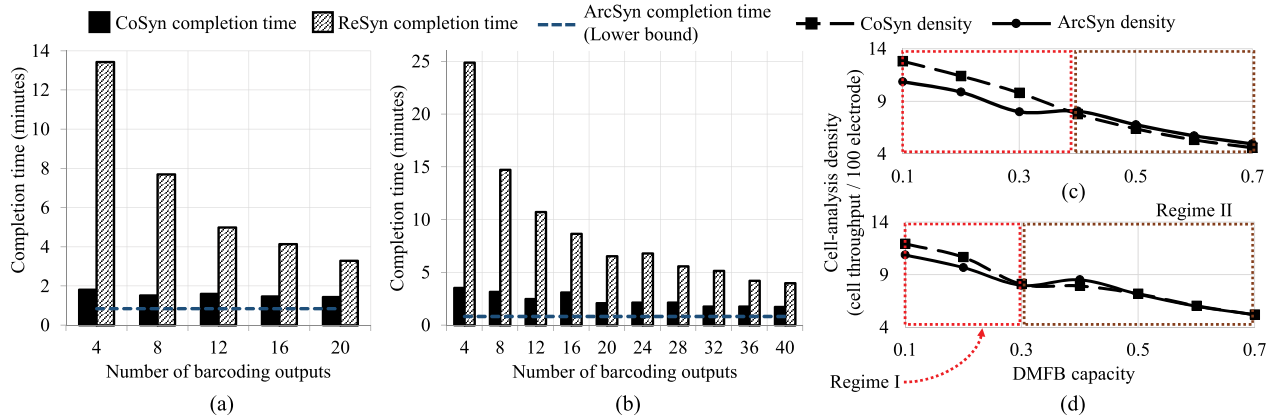
mapping between syringe pumps and DMFB ports as in Fig. 21(b); and 2) algorithmic baseline (ReSyn), in which resource allocation is initially performed for each microfluidic domain separately. However, we must ensure that the system behavior at the boundary between the two domains is deterministic—the synthesis tool for the DMFB must be aware of the order of the barcoding droplets generated from the valve-based crossbar. The only way to meet this constraint is to disallow pipelining in the valve-based system; thus, an upper bound on the number of barcoding droplets that can be processed simultaneously is equal to  $m$ . Since we consider a large number of cells, we divide the cells into batches, each of a maximum size of  $m$  cells, such that ReSyn executes them iteratively.

1) *Performance Evaluation*: We evaluate the performance of CoSyn, ArcSyn, and ReSyn in terms of the total completion time for the protocol, measured in minutes (we assume  $T_f = T_d = 0.2$  s). The assumption of  $T_d = 0.2$  s is based on the fact that a DMFB can flexibly function at any rate in the range from 1 to 100 Hz, as described in Section V-C3. Therefore, a rate of 5 Hz is a valid choice. On the other side,  $T_f = 0.2$  s is associated with a previous flow-based design that was operated using a pneumatic transporter with a pumping rate of 50 Hz. The reported flow speed in this design is in the range of 10–20 mm/s [92]. In our example, a transposer with a length of 4 mm with the proper channel configurations (PDMS material and diameter) can lead to a flow speed of  $4 \text{ mm}/0.2 \text{ s} = 20 \text{ mm/s}$  [14].

We also fix the number of input cells to 100, and we consider 20 and 40 barcoding inputs (or cell types). To ensure that this evaluation is independent of the platform architecture, the results were obtained using a DMFB with no resource constraints. The selected number of cells is based on a design that was introduced earlier in [34]. Usually, a cell suspension (an experiment session) using a microfluidic prototype includes a number of cells in the range of 100–1000 cells. A larger number of cells per suspension requires more effort during cell isolation and sorting.

Fig. 26(a)–(b) compares the three synthesis frameworks in terms of completion times. ReSyn leads to the highest completion times due to the loose coordination between the DMFB and the valve-based crossbar. The completion time of CoSyn is close to the lower bound, which is obtained using ArcSyn. ArcSyn uses the maximum number of barcoding outputs due to the one-to-one mapping between the barcoding inputs and outputs. Hence, these results indicate that pipelined valve-based routing and the coordination between the components of CoSyn play a key role in increasing cell-analysis throughput.

2) *Design-Quality Assessment*: We also evaluate the quality of the designs generated by CoSyn in terms of the number



**Fig. 26. Comparison between CoSyn, ArcSyn, and ReSyn: (a) completion time ( $T$ ) using 20 barcoding inputs; (b)  $T$  for 40 barcoding inputs; (c) cell-analysis density ( $D$ ) using four barcoding outputs; and (d)  $D$  using eight barcoding outputs.**

of single-cell experiments that can be completed for a given time limit and the given number of DMFB resources. In addition, we also quantify the fraction of input cells that can be processed simultaneously by the given set of DMFB resources. Our objective here is to investigate the conditions under which CoSyn is effective. Therefore, we introduce the following terms.

- Cell-analysis density: The number of cells (samples) that completed analysis during a specific window of time (cell throughput), using a given array of electrodes. The time window is set to be a minute and the size of the array is equal to 100 electrodes.
- DMFB capacity: A real number  $z \in [0, 1]$  that provides the fraction of input cells that can be processed simultaneously using DMFB resources. For example, a capacity of 1 indicates that there are sufficient resources to process all the cells simultaneously. On the other hand, a capacity of 0.5 means that the existing resources are sufficient for simultaneously processing only half of the cells.

We investigate the design quality for valve-based crossbars by evaluating the cell-analysis density of CoSyn and ArcSyn. We simulate the execution of 50 cells using four barcoding outputs [Fig. 26(c)] and eight barcoding outputs [Fig. 26(d)]. The density values are computed while the capacity is varied. By comparing the density values for CoSyn and ArcSyn, we observe two regimes: 1) regime I in which the cell-analysis density of CoSyn is higher, i.e., it is more effective; and 2) regime II in which the density of CoSyn is less than or equal to the density of ArcSyn. Regime I highlights the fact that CoSyn efficiently exploits valve-based barcoding, and the power of valve-based pipelining is evident when the DMFB resources are limited. On the other hand, the overprovisioning of resources leads to regime II, where a lower cell-analysis density is reported. Finally, we note that regime I shrinks

as we increase the number of barcoding outputs; this is expected since CoSyn is more effective in the realistic case of a limited number of barcoding interfaces. This study has been confirmed using different settings corresponding to the number of cells and the number of barcoding inputs and outputs.

## VII. CONCLUSION AND FUTURE OUTLOOK

Digital microfluidics has the potential to change the way contemporary microbiology research is performed. To foster the adoption of digital microfluidics in microbiology and biochemistry, we have introduced a new synthesis methodology that helps bridge the gap between biochips designed by engineers and actual use of these chips by biologists. Motivated by real-life benchtop studies, the proposed synthesis methodology exploits cyber-physical integration and introduces truly enabling designs for large-scale biomolecular protocols. This paper is a call to action for researchers in cyber-physical microfluidic systems and biochip design automation to engage in the transition toward realistic microbiology-on-chip.

As the complexity of biomolecular protocols increases, more demands are placed on biochip designers to meet precision requirements while taking into consideration platform costs. The step toward bridging the gap between microfluidics and microbiology is to address the first step, i.e., specification-driven experiment design, in the workflow shown in Fig. 4. For example, chromatin immunoprecipitation coupled with high-throughput DNA sequencing (ChIP-seq) can offer high-resolution, genome-wide analysis of DNA-protein interactions. Ideally, a designer can plan an experiment to investigate the interaction between a certain DNA and all possible proteins activated by neighboring genes. However, this plan requires a significant amount of chemicals that can be used for crosslinking all



possible protein molecules to DNA. In addition, the duration of crosslinking and the concentration of the chemicals are variables that depend on the experiment environment. These complications (and others) need to be thoroughly studied and co-optimized before starting the actual synthesis. ■

## Acknowledgment

The authors would like to thank Prof. K. Scott and Prof. M. Pajic from Duke University, Durham, NC, USA; and Prof. U. Schlichtmann from Technical University of Munich, Munich, Germany, for their contributions to various aspects of the work reported in this paper.

## REFERENCES

- [1] G. M. Whitesides, "The origins and the future of microfluidics," *Nature*, vol. 442, no. 7101, pp. 368–373, 2006.
- [2] M. Karle *et al.*, "Microfluidic solutions enabling continuous processing and monitoring of biological samples: A review," *Anal. Chim. Acta*, vol. 929, pp. 1–22, Jul. 2016.
- [3] Yole Développement, "Microfluidic applications in the pharmaceutical, life sciences, in-vitro diagnostic, and medical device markets," 2015. [Online]. Available: [http://www.yole.fr/Microfluidic\\_Applications\\_Players.aspx](http://www.yole.fr/Microfluidic_Applications_Players.aspx)
- [4] M. G. Pollack, V. K. Pamula, V. Srinivasan, and A. E. Eckhardt, "Applications of electrowetting-based digital microfluidics in clinical diagnostics," *Expert Rev. Molecular Diagnostics*, vol. 11, no. 4, pp. 393–407, 2011.
- [5] D. G. Rackus, M. H. Shamsi, and A. R. Wheeler, "Electrochemistry, biosensors and microfluidics: A convergence of fields," *Chem. Soc. Rev.*, vol. 44, no. 15, pp. 5320–5340, 2015.
- [6] T. Thorsen, S. J. Maerkl, and S. R. Quake, "Microfluidic large-scale integration," *Science*, vol. 298, no. 5593, pp. 580–584, 2002.
- [7] E. K. Sackmann, A. L. Fulton, and D. J. Beebe, "The present and future role of microfluidics in biomedical research," *Nature*, vol. 507, no. 7491, pp. 181–189, 2014.
- [8] Fluidigm Corporation. Accessed: May 27, 2017. [Online]. Available: <https://www.fluidigm.com/systems>
- [9] R. Silva *et al.*, "A reconfigurable continuous-flow fluidic routing fabric using a modular, scalable primitive," *Lab Chip*, 2016.
- [10] L. M. Fidalgo and S. J. Maerkl, "A software-programmable microfluidic device for automated biology," *Lab Chip*, vol. 11, no. 9, pp. 1612–1619, 2011.
- [11] J. Kim, A. M. Stockton, E. C. Jensen, and R. A. Mathies, "Pneumatically actuated microvalve circuits for programmable automation of chemical and biochemical analysis," *Lab Chip*, vol. 16, no. 5, pp. 812–819, 2016.
- [12] Q. Wang *et al.*, "Hamming-distance-based valve-switching optimization for control-layer multiplexing in flow-based microfluidic biochips," in *Proc. IEEE/ACM Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2017, pp. 524–529.
- [13] C. Liu, B. Li, B. B. Bhattacharya, K. Chakrabarty, T.-Y. Ho, and U. Schlichtmann, "Testing microfluidic fully programmable valve arrays (FPVAs)," in *Proc. IEEE/ACM Design Autom. Test Europe (DATE) Conf.*, Mar. 2017, pp. 91–96.
- [14] M. Ibrahim, A. Sridhar, K. Chakrabarty, and U. Schlichtmann, "Sortex: Efficient timing-driven synthesis of reconfigurable flow-based biochips for scalable single-cell screening," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, to be published.
- [15] Illumina, "Illumina NeoPrep Library Prep System." Accessed: May 27, 2017. [Online]. Available: <http://www.illumina.com/systems/neo-prep-library-system.html/>
- [16] GenMark Dx, "GenMark ePlex System." Accessed: May 27, 2017. [Online]. Available: <https://www.genmarkdx.com/solutions/systems/eplex-system/>
- [17] Baebies, "Seeker platform for newborn screening." Accessed: May 27, 2017. [Online]. Available: <http://baebies.com/products/seeker/>
- [18] R. Sista *et al.*, "Development of a digital microfluidic platform for point of care testing," *Lab Chip*, vol. 8, no. 12, pp. 2091–2104, 2008.
- [19] 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.
- [20] D. J. Boles *et al.*, "Droplet-based pyrosequencing using digital microfluidics," *Anal. Chem.*, vol. 83, no. 22, pp. 8439–8447, 2011.
- [21] Y. Zhao, S. K. Chung, U.-C. Yi, and S. K. Cho, "Droplet manipulation and microparticle sampling on perforated microfilter membranes," *J. Micromech. Microeng.*, vol. 18, no. 2, p. 025030, 2008.
- [22] M. G. Pollack, A. D. Shenderov, and R. B. Fair, "Electrowetting-based actuation of droplets for integrated microfluidics," *Lab Chip*, vol. 2, no. 2, pp. 96–101, 2002.
- [23] S. K. Cho, H. Moon, and C.-J. Kim, "Creating, transporting, cutting, and merging liquid droplets by electrowetting-based actuation for digital microfluidic circuits," *J. Microelectromech. Syst.*, vol. 12, no. 1, pp. 70–80, Feb. 2003.
- [24] R. B. Fair, "Digital microfluidics: Is a true lab-on-a-chip possible?" *Microfluidics Nanofluidics*, vol. 3, no. 3, pp. 245–281, Mar. 2007.
- [25] B.-N. Hsu, "Accelerated sepsis diagnosis by seamless integration of nucleic acid purification and detection," Ph.D. dissertation, Duke Univ., Durham, NC, USA, 2014.
- [26] L. Chen, "Sparse sample detection using magnetic bead manipulation on a digital microfluidic device," Ph.D. dissertation, Duke Univ., Durham, NC, USA, 2016.
- [27] A. C. Madison, "Scalable genome engineering in electrowetting on dielectric digital microfluidic systems," Ph.D. dissertation, Duke Univ., Durham, NC, USA, 2015.
- [28] K. Chakrabarty and T. Xu, *Digital Microfluidic Biochips: Design Automation and Optimization*. Boca Raton, FL, USA: CRC Press, 2010.
- [29] P. Pop, M. Alistar, E. Stuart, and J. Madsen, "Design methodology for digital microfluidic biochips," in *Fault-Tolerant Digital Microfluidic Biochips*. New York, NY, USA: Springer-Verlag, 2016, pp. 13–28.
- [30] C. D. Chin *et al.*, "Microfluidics-based diagnostics of infectious diseases in the developing world," *Nature Med.*, vol. 17, no. 8, pp. 1015–1019, 2011.
- [31] M. Ibrahim, K. Chakrabarty, and K. Scott, "Synthesis of cyberphysical digital-microfluidic biochips for real-time quantitative analysis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 5, pp. 733–746, May 2017.
- [32] M. Ibrahim and K. Chakrabarty, "Digital-microfluidic biochips for quantitative analysis: Bridging the gap between microfluidics and microbiology," in *Proc. IEEE/ACM Design Autom. Test Europe (DATE) Conf.*, Mar. 2017, pp. 1787–1792.
- [33] M. J. Jebraïl *et al.*, "A solvent replenishment solution for managing evaporation of biochemical reactions in air-matrix digital microfluidics devices," *Lab Chip*, vol. 15, no. 1, pp. 151–158, 2015.
- [34] A. Rival *et al.*, "An EWOD-based microfluidic chip for single-cell isolation, mRNA purification and subsequent multiplex qPCR," *Lab Chip*, vol. 14, no. 19, pp. 3739–3749, 2014.
- [35] E. Brouzes *et al.*, "Droplet microfluidic technology for single-cell high-throughput screening," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 34, pp. 14195–14200, 2009.
- [36] L. Malic, D. Brassard, T. Veres, and M. Tabrizian, "Integration and detection of biochemical assays in digital microfluidic LOC devices," *Lab Chip*, vol. 10, no. 4, pp. 418–431, 2010.
- [37] B. S. Wheeler, B. T. Ruderman, H. F. Willard, and K. C. Scott, "Uncoupling of genomic and epigenetic signals in the maintenance and inheritance of heterochromatin domains in fission yeast," *Genetics*, vol. 190, no. 2, pp. 549–557, 2012.
- [38] J. C. Contreras-Naranjo, Q. Wei, and A. Ozcan, "Mobile phone-based microscopy, sensing, and diagnostics," *IEEE J. Sel. Topics Quantum Electron.*, vol. 22, no. 3, pp. 1–14, Mar. 2016.
- [39] A. Ozcan and E. McLeod, "Lensless imaging and sensing," *Annu. Rev. Biomed. Eng.*, vol. 18, no. 1, pp. 77–102, 2016.
- [40] M. Ibrahim, Z. Li, and K. Chakrabarty, "Advances in design automation techniques for digital-microfluidic biochips," in *Formal Modeling and Verification of Cyber-Physical Systems*. New York, NY, USA: Springer-Verlag, 2015, pp. 190–223.
- [41] F. Su and K. Chakrabarty, "Architectural-level synthesis of digital microfluidics-based biochips," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2004, pp. 223–228.
- [42] F. Su and K. Chakrabarty, "Module placement for fault-tolerant microfluidics-based biochips," *ACM Trans. Design Autom. Electron. Syst.*, vol. 11, no. 3, pp. 682–710, 2004.
- [43] F. Su and K. Chakrabarty, "High-level synthesis of digital microfluidic biochips," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 3, no. 4, p. 1, 2008.
- [44] F. Su, W. Hwang, and K. Chakrabarty, "Droplet routing in the synthesis of digital

## Ibrahim and Chakrabarty: Cyber-Physical Digital-Microfluidic Biochips

- microfluidic biochips," in *Proc. IEEE/ACM Design Autom. Test Europe (DATE) Conf.*, Mar. 2006, pp. 1–6.
- [45] T. Xu and K. Chakrabarty, "Integrated droplet routing in the synthesis of microfluidic biochips," in *Proc. IEEE/ACM Design Autom. Conf. (DAC)*, Jun. 2007, pp. 948–953.
- [46] P.-H. Yuh, C.-L. Yang, and Y.-W. Chang, "BioRoute: A network-flow-based routing algorithm for the synthesis of digital microfluidic biochips," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Mar. 2007, pp. 752–757.
- [47] M. Cho and D. Z. Pan, "A high-performance droplet routing algorithm for digital microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr.*, vol. 27, no. 10, pp. 1714–1724, Oct. 2008.
- [48] D. Grissom and P. Brisk, "A field-programmable pin-constrained digital microfluidic biochip," in *Proc. IEEE/ACM Design Autom. Conf. (DAC)*, May 2013, p. 46.
- [49] 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 Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 12, pp. 1786–1799, Dec. 2011.
- [50] T. Xu, K. Chakrabarty, and V. K. Pamula, "Defect-tolerant design and optimization of a digital microfluidic biochip for protein crystallization," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 4, pp. 552–565, Apr. 2010.
- [51] Y. Luo, K. Chakrabarty, and T. Y. Ho, "Error recovery in cyberphysical digital microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 1, pp. 59–72, Jan. 2013.
- [52] M. Alistar, P. Pop, and J. Madsen, "Redundancy optimization for error recovery in digital microfluidic biochips," *Design Autom. Embedded Syst.*, vol. 19, nos. 1–2, pp. 129–159, 2015.
- [53] C. Jaress, P. Brisk, and D. Grissom, "Rapid online fault recovery for cyber-physical digital microfluidic biochips," in *Proc. IEEE VLSI Test Symp. (VTS)*, Apr. 2015, pp. 1–6.
- [54] M. Ibrahim and K. Chakrabarty, "Error recovery in digital microfluidics for personalized medicine," in *Proc. IEEE/ACM Design Autom. Test Europe (DATE) Conf.*, Mar. 2015, pp. 247–252.
- [55] S. Poddar, S. Ghoshal, K. Chakrabarty, and B. B. Bhattacharya, "Error-correcting sample preparation with cyberphysical digital microfluidic lab-on-chip," *ACM Trans. Design Autom. Electron. Syst.*, vol. 22, no. 1, p. 2:1–29, 2016.
- [56] Y. Luo, B. B. Bhattacharya, T. Y. Ho, and K. Chakrabarty, "Design and optimization of a cyberphysical digital-microfluidic biochip for the polymerase chain reaction," *IEEE Trans. Comput.-Aided Design Integr.*, vol. 34, no. 1, pp. 29–42, Jan. 2015.
- [57] J. Gao *et al.*, "An intelligent digital microfluidic system with fuzzy-enhanced feedback for multi-droplet manipulation," *Lab Chip*, vol. 13, no. 3, pp. 443–451, 2013.
- [58] K. Hu, M. Ibrahim, L. Chen, Z. Li, K. Chakrabarty, and R. Fair, "Experimental demonstration of error recovery in an integrated cyberphysical digital-microfluidic platform," in *Proc. IEEE Biomed. Circuits Syst. (BioCAS) Conf.*, Oct. 2015, pp. 1–4.
- [59] A. R. Wu *et al.*, "Automated microfluidic chromatin immunoprecipitation from 2,000 cells," *Lab Chip*, vol. 9, no. 10, pp. 1365–1370, 2009.
- [60] R. Zilionis *et al.*, "Single-cell barcoding and sequencing using droplet microfluidics," *Nature Protocols*, vol. 12, no. 1, pp. 44–73, 2017.
- [61] D. Sims, I. Sudbery, N. E. Iltott, A. Heger, and C. P. Ponting, "Sequencing depth and coverage: Key considerations in genomic analyses," *Nature Rev. Genet.*, vol. 15, no. 2, pp. 121–132, 2014.
- [62] A. M. Klein *et al.*, "Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells," *Cell*, vol. 161, no. 5, pp. 1187–1201, 2015.
- [63] T. Xu and K. Chakrabarty, "Broadcast electrode-addressing for pin-constrained multi-functional digital microfluidic biochips," in *Proc. IEEE/ACM Design Autom. Conf. (DAC)*, Jun. 2008, pp. 173–178.
- [64] P. D. S. Bortz and B. R. Wamhoff, "Chromatin immunoprecipitation (ChIP): Revisiting the efficacy of sample preparation, sonication, quantification of sheared DNA, and analysis via PCR," *PLoS ONE*, vol. 6, no. 10, p. e26015, 2011.
- [65] S. Hoscic, S. K. Murthy, and A. N. Koppes, "Microfluidic sample preparation for single cell analysis," *Anal. Chem.*, vol. 88, no. 1, pp. 354–380, 2015.
- [66] V. Srinivasan, V. K. Pamula, and R. B. Fair, "Droplet-based microfluidic lab-on-a-chip for glucose detection," *Anal. Chim. Acta*, vol. 507, no. 1, pp. 145–150, 2004.
- [67] M. Ibrahim, K. Chakrabarty, and K. Scott, "Integrated and real-time quantitative analysis using cyberphysical digital-microfluidic biochips," in *Proc. IEEE/ACM Design Autom. Test Europe (DATE) Conf.*, Mar. 2016, pp. 630–635.
- [68] E. Wulff-Burchfield *et al.*, "Microfluidic platform versus conventional real-time polymerase chain reaction for the detection of mycoplasma pneumoniae in respiratory specimens," *Diagnostic Microbiol. Infectious Disease*, vol. 67, no. 1, pp. 22–29, 2010.
- [69] Y. Xie and W. Wolf, "Allocation and scheduling of conditional task graph in hardware/software co-synthesis," in *Proc. IEEE/ACM Design Autom. Test Europe (DATE) Conf.*, Mar. 2001, pp. 620–625.
- [70] P. Pop, P. Eles, and Z. Peng, *Analysis and Synthesis of Distributed Real-Time Embedded Systems*. New York, NY, USA: Springer-Verlag, 2013.
- [71] H. Norian, R. M. Field, I. Kymissis, and K. L. Shepard, "An integrated CMOS quantitative-polymerase-chain-reaction lab-on-chip for point-of-care diagnostics," *Lab Chip*, vol. 14, no. 20, pp. 4076–4084, 2014.
- [72] A. G. Papatthanasasiou and A. G. Boudouvis, "Manifestation of the connection between dielectric breakdown strength and contact angle saturation in electrowetting," *Appl. Phys. Lett.*, vol. 86, no. 16, p. 164102, 2005.
- [73] R. Adler, I. Schaefer, M. Trapp, and A. Poetsch-Heffter, "Component-based modeling and verification of dynamic adaptation in safety-critical embedded systems," *ACM Trans. Embedded Comput. Syst.*, vol. 10, no. 2, p. 20, 2010.
- [74] L. A. Roza Duque, J. M. M. Diaz, and C. Yang, "Improving MPSoC reliability through adapting runtime task schedule based on time-correlated fault behavior," in *Proc. IEEE/ACM Design Autom. Test Europe (DATE)*, Mar. 2015, pp. 818–823.
- [75] M. Ibrahim, C. Boswell, K. Chakrabarty, K. Scott, and M. Pajic, "A real-time digital-microfluidic platform for epigenetics," in *Proc. Int. Conf. Compilers Archit. Synthesis Embedded Syst. (CASES)*, vol. 10, 2016, pp. 1–10.
- [76] L. T. X. Phan, I. Lee, and O. Sokolsky, "Compositional analysis of multi-mode systems," in *Proc. IEEE Euromicro Conf. Real-Time Syst. (ECRTS)*, Jul. 2010, pp. 197–206.
- [77] S. Baruah, M. Bertogna, and G. Buttazzo, *Multiprocessor Scheduling for Real-Time Systems*. New York, NY, USA: Springer-Verlag, 2015.
- [78] V. Bonifaci, A. Marchetti-Spaccamela, S. Stiller, and A. Wiese, "Feasibility analysis in the sporadic DAG task model," in *Proc. IEEE Euromicro Conf. Real-Time Syst. (ECRTS)*, Jul. 2013, pp. 225–233.
- [79] 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, 2003.
- [80] A. Saifullah, D. Ferry, J. Li, K. Agrawal, C. Lu, and C. D. Gill, "Parallel real-time scheduling of DAGs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 12, pp. 3242–3252, Dec. 2014.
- [81] A. K. Mok, "Fundamental design problems of distributed systems for the hard-real-time environment," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 1983.
- [82] 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, Dec. 2015.
- [83] M. Ibrahim, K. Chakrabarty, and U. Schlichtmann, "CoSyn: Efficient single-cell analysis using a hybrid microfluidic platform," in *Proc. IEEE/ACM Design Autom. Test Europe (DATE) Conf.*, Mar. 2017, pp. 1673–1678.
- [84] A. Saadatpour, S. Lai, G. Guo, and G.-C. Yuan, "Single-cell analysis in cancer genomics," *Trends Genet.*, vol. 31, no. 10, pp. 576–586, 2015.
- [85] N. Shembekar, C. Chaipan, R. Utharala, and C. A. Merten, "Droplet-based microfluidics in drug discovery, transcriptomics and high-throughput molecular genetics," *Lab Chip*, vol. 16, no. 8, pp. 1314–1331, 2016.
- [86] M. J. Jebrail *et al.*, "World-to-digital-microfluidic interface enabling extraction and purification of RNA from human whole blood," *Anal. Chem.*, vol. 86, no. 8, pp. 3856–3862, 2014.
- [87] S. C. Shih *et al.*, "A droplet-to-digital (D2D) microfluidic device for single cell assays," *Lab Chip*, vol. 15, no. 1, pp. 225–236, 2015.
- [88] M. A. Murran and H. Najjaran, "Capacitance-based droplet position estimator for digital microfluidic devices," *Lab Chip*, vol. 12, no. 11, pp. 2053–2059, 2012.
- [89] A. R. Wu *et al.*, "High throughput automated chromatin immunoprecipitation as a platform for drug screening and antibody validation," *Lab Chip*, vol. 12, no. 12, pp. 2190–2198, 2012.
- [90] M. Ibrahim, K. Chakrabarty, and U. Schlichtmann, "CoSyn: Efficient single-cell analysis using a hybrid microfluidic platform," Duke Univ., Durham, NC, USA, Tech. Rep. 10161/13011. [Online]. Available: <http://hdl.handle.net/10161/13011>
- [91] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2001.
- [92] Y. Lim, A. Kouzani, and W. Duan, "Lab-on-a-chip: A component view," *Microsyst. Technol.*, vol. 16, no. 12, pp. 1995–2015, 2010.

## ABOUT THE AUTHORS

**Mohamed Ibrahim** (Student Member, IEEE) received the B.Sc. (honors) degree in electrical engineering and the M.Sc. degree from Ain Shams University, Cairo, Egypt, in 2010 and 2013, respectively. Currently, he is working toward the Ph.D. degree at the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA.



He was a design-for-test intern with Intel Corporation, Santa Clara, CA, USA, and Intel Corporation, Austin, TX, USA. He was also a visiting scholar with the Technical University of Munich, Munich, Germany, and the University of Bremen, Bremen, Germany. His research interests include design automation of microfluidic systems for biomolecular analysis, security of microfluidic biochips, and design-for-test of emerging technologies. He has published over 20 papers on these topics in journals and refereed conference proceedings.

Mr. Ibrahim is a recipient of the 2017 Postdoc Mobility Travel award from the Technical University of Munich and the Best Paper award at the 2017 IEEE/ACM Design, Automation, and Test in Europe (DATE) Conference. He is a student member of the Association for Computing Machinery (ACM).

**Krishnendu Chakrabarty** (Fellow, IEEE) received the B.Tech. degree from the Indian Institute of Technology, Kharagpur, India, in 1990 and the M.S.E. and Ph.D. degrees from the University of Michigan, Ann Arbor, MI, USA, in 1992 and 1995, respectively.



He is currently the William H. Younger Distinguished Professor and Chair in the Department of Electrical and Computer Engineering and Professor of Computer Science at Duke University, Durham, NC, USA. He has held Visiting Professor positions at the University of Tokyo, Tokyo, Japan and the Nara Institute of Science and Technology (NAIST), Nara, Ikoma, Japan, and Visiting Chair Professor positions at Tsinghua University, Beijing, China and the National Cheng Kung University, Tainan, Taiwan. His current research projects include: testing and design-for-testability of integrated circuits and systems; digital microfluidics, biochips, and cyber-physical systems; data analytics for fault diagnosis, failure

prediction, anomaly detection, and hardware security; and neuromorphic computing systems. He has authored 20 books on these topics (with one translated into Chinese), published over 630 papers in journals and refereed conference proceedings, and given over 270 invited, keynote, and plenary talks. He has also presented 60 tutorials at major international conferences. He holds ten U.S. patents, with several patents pending.

Prof. Chakrabarty is a Fellow of the Association for Computing Machinery (ACM) and a Golden Core Member of the IEEE Computer Society. He is a recipient of the National Science Foundation CAREER award, the U.S. Office of Naval Research Young Investigator award, the Humboldt Research Award from the Alexander von Humboldt Foundation, Germany, the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN Donald O. Pederson Best Paper Award (2015), the *ACM Transactions on Design Automation of Electronic Systems* Best Paper Award (2016), and over a dozen best paper awards at major conferences. He is also a recipient of the IEEE Computer Society Technical Achievement Award (2015), the IEEE Circuits and Systems Society Charles A. Desoer Technical Achievement Award (2017), and the Distinguished Alumnus Award from the Indian Institute of Technology, Kharagpur (2014). He is a Research Ambassador of the University of Bremen (Germany) and a Hans Fischer Senior Fellow (named after Nobel Laureate Prof. Hans Fischer) at the Institute for Advanced Study, Technical University of Munich, Germany. He is a De Tao Master of Emerging Information Technologies, DeTao Masters Academy of the DeTao Group in China. He was a 2009 Invitational Fellow of the Japan Society for the Promotion of Science (JSPS). He is a recipient of the 2008 Duke University Graduate School Dean's Award for excellence in mentoring, and the 2010 Capers and Marion McDonald Award for Excellence in Mentoring and Advising, Pratt School of Engineering, Duke University. He has served as a Distinguished Visitor of the IEEE Computer Society (2005–2007, 2010–2012), a Distinguished Lecturer of the IEEE Circuits and Systems Society (2006–2007, 2012–2013), and an ACM Distinguished Speaker (2008–2016). He served as the Editor-in-Chief of IEEE DESIGN & TEST OF COMPUTERS in 2010–2012 and *ACM Journal on Emerging Technologies in Computing Systems* in 2010–2015. Currently he serves as the Editor-in-Chief of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS. He is also an Associate Editor of the IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS, the IEEE TRANSACTIONS ON MULTISCALE COMPUTING SYSTEMS, and the *ACM Transactions on Design Automation of Electronic Systems*.