UCR Dilution Framework

July 11, 2015

1 Dilution Framework

The Dilution Framework is the combination of several dilution algorithms that have been published by various institutions. Each algorithm outputs a mixing tree formatted as a dag to be used by other applications

1.1 Output formats

There are several output options that can be chosen.

1. Digital Microfluidics Instructions
2. Continuous Flow Microfluidics Instructions
3. Dilution Generator File
4. Doty Graph

1.2 Output Command Arguments

To specify the type of output you want, first specify the desired output type then the file name. To specify the console as output use "CONSOLE" as file name. Additionally the file name cannot have any spaces. To run use the following Commands:

DIGITAL FileName <Algorithm Command Line Arguments>
FLOW FileName <Algorithm Command Line Arguments>
DAGGEN FileName <Algorithm Command Line Arguments>
DOTY FileName <Algorithm Command Line Arguments>
ALL FileName <Algorithm Command Line Arguments>

2 Algorithms

List of working Algorithms in generator. A short description of each algorithm can be seen below.

1. MinMix
   $< OutputCommandArguments >$ MINMIX CV_1 CV_2 ... $CName_1$ $CName_2$ ...
2. MTC
   $< OutputCommandArguments >$ MTC denominator $numerator_1$ $numerator_2$ ...
3. Remia
   $< OutputCommandArguments >$ REMIA numerator denominator
4. CoDos
   Priced Reageants:
   $< OutputCommandArguments >$ CODOS PRICED CV_{R1} Wt_{R1} CV_{R2} Wt_{R2} ...
   NON-Priced Reageants:
   $< OutputCommandArguments >$ CODOS CV_{R1} CV_{R2} ...
5. NRT_ISI
   $< OutputCommandArguments >$ NRT_ISI CV_1 CV_2 ...
6. Linear Gradient
   $< OutputCommandArguments >$ ISI_NCKU numerator differnceBetweenSamples n^2(denominator) numSamples
7. IDMA
   $< OutputCommandArguments >$ IDMA numOps tolerance desiredConcentration
8. DMRW
   $< OutputCommandArguments >$ DMRW numOps tolerance desiredConcentration
9. GRIFFITH
   $< OutputCommandArguments >$ GRIFFITH numOps tolerance desiredConcentration

3 Algorithm Descriptions

3.1 Min Mix

Inputs: Multiple Reagents
Outputs: Single Composite Reagent
Optimization: None

The concentration values(CV) must be integers. Although it is not required, for the most accurate output, the sum of the CVs should be divisible by powers of 2. Optionally, you may also specify the name (CName) for the concentration value that was passed in. MinMix is implemented as presented by Thies et al. [9]

3.2 MTC

Inputs: Single Reagent
Output: Multiple Diluted Reagents as specified.
Optimization: Minimum number mix-split steps to generate multiple diluent.

Inputs are the denominator and each numerator that is smaller than the denominator. The denominator **must** be divisible by power of 2. This algorithm follows the specification in Mitra et al[6]. The traveling salesmen problem is solved using the software available here: http://www.akira.ruc.dk/ keld/research/LKH/.

3.3 Remia

Inputs: Single Reagent
Output: Single Diluted Reagent
Optimization: Reactant Minimization

The inputs are the numerator and dominator for the final diluent. This algorithm follows the Remia implementation as specified by Huang et al. [4].

3.4 CODOS

Inputs: Multiple Reagents
Outputs: Single Composite Reagent
Optimization: Reactant Minimization, Price Minimization

The CVs for this algorithm must be integers. Additionally the sum of the CVs **must** be divisible by a power of 2. This algorithm also allows specification of price of the reagent. This algorithm follows the specification as presented in [5].

3.5 NRT_ISI

Inputs: Multiple CVs of the same reagent
Output: Multiple Diluted Reagents at desired CV.
Optimization: Optimized for Mix-Split steps and waste droplets.

The CV's need to add up to a power of 2 number. The implementation follow the specification in [1].

3.6 Linear Gradient

Inputs: numerator differnceBetweenSamples n^2(denominator) numSamples
Output: Series of dilutes starting at the specified numerator.
Optimization: Optimized for Mix-Split steps and waste droplets for a linear series generation.

Creates a series of diluted reagents starting at numerator and then outputing the every sample that is the difference apart. The implementatoin follows the specification from [2].

An example input of "100 3 7 5" would generate a mixing tree for: $\frac{100}{256}, \frac{103}{256}, \frac{106}{256}, \frac{109}{256}, \frac{112}{256}$.

3.7 IDMA

Inputs: Single Reagent
Output: Single Diluent
Optimization:

NumOps specifies the maximum number of operations(Mix-Split) steps that are allowed. The tolerance value bounds the rounding errors that occur from splitting droplets. The implementation follows the specification in [7]

3.8 DMRW

Inputs: Single Reagent
Output: Single Diluent
Optimization: Dilution and Mixing with Reduced Wastage

NumOps specifies the maximum number of operations(Mix-Split) steps that are allowed. The tolerance value bounds the rounding errors that occur from splitting droplets. The implementation follows the specification in [8]

3.9 GRIFFITH

Inputs: Single Reagent
Output: Single Diluent
Optimization:

NumOps specifies the maximum number of operations(Mix-Split) steps that are allowed. The tolerance value bounds the rounding errors that occur from splitting droplets. The implementation follows the specification in [3]

References

[1] S. Bhattacharjee, A. Banerjee, and B. B. Bhattacharya. Multiple Dilution Sample Preparation Using Digital Microfluidic Biochips. *2012 International Symposium on Electronic System Design (ISED)*, pages 188–192, Dec. 2012.

[2] S. Bhattacharjee, A. Banerjee, T.-Y. Ho, K. Chakrabarty, and B. B. Bhattacharya. On Producing Linear Dilution Gradient of a Sample with a Digital Microfluidic Biochip. *2013 International Symposium on Electronic System Design*, pages 77–81, Dec. 2013.

[3] E. Griffith, S. Akella, and M. Goldberg. Performance Characterization of a Reconfigurable Planar-Array Digital Microfluidic System. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(2):345–357, Feb. 2006.

[4] J.-D. Huang, C.-H. Liu, and T.-W. Chiang. Reactant minimization during sample preparation on digital microfluidic biochips using skewed mixing trees. *Proceedings of the International Conference on Computer-Aided Design - ICCAD '12*, 2012.

[5] C.-H. Liu, H.-H. Chang, T.-C. Liang, and J.-D. Huang. Sample preparation for many-reactant bioassay on DMFBs using common dilution operation sharing. *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 615–621, Nov. 2013.

[6] D. Mitra, S. Roy, K. Chakrabarty, and B. B. Bhattacharya. On-Chip Sample Preparation with Multiple Dilutions Using Digital Microfluidics. *2012 IEEE Computer Society Annual Symposium on VLSI*, pages 314–319.

[7] S. Roy, B. B. Bhattacharya, and K. Chakrabarty. Waste-aware dilution and mixing of biochemical samples with digital microfluidic biochips. *2011 Design, Automation & Test in Europe*, pages 1–6, 2011.

[8] S. Roy, S. Member, and B. B. Bhattacharya. Optimization of Dilution and Mixing of Biochemical Samples Using Digital Microfluidic Biochips. 29(11):1696–1708, 2010.

[9] W. Thies, J. P. Urbanski, T. Thorsen, and S. Amarasinghe. Abstraction layers for scalable microfluidic biocomputing. *Natural Computing*, 7(2):255–275, 5 2007.