Supplementary Information for

Finding the optimal design of a passive microfluidic mixer

Junchao Wang,1,2* Naiyin Zhang,3
Jin Chen,1 Victor G. J. Rodgers,2
Philip Brisk,4 and William H. Grover2*

1Key Laboratory of RF Circuits and Systems, Ministry of Education,
and Zhejiang Provincial Laboratory of Integrated Circuit Design,
Hangzhou Dianzi University, China.
2Department of Bioengineering, University of California Riverside,
Riverside, CA, USA.
3College of Life Information Science and Instrument Engineering,
Hangzhou Dianzi University, China.
4Department of Computer Science and Engineering,
University of California Riverside, Riverside, CA, USA.

1 Description of supplementary files

1.1 Video files

• s1.mp4 is the concentration profiles of 0 to 200 generations of NSGA-II mixer designs.
• s2.mp4 is the pressure profiles of 0 to 200 generations of NSGA-II mixer designs.
• s3.mp4 is the velocity magnitude profiles of 0 to 200 generations of NSGA-II mixer designs.

1.2 Code to generate random mixer designs

The file code.7z contains the MATLAB script we used to generate 6069 random mixer designs.

Program entrance
1.3 Guideline to implement NSGA-II for specific applications

We used a NSGA-II package implemented in MATLAB from github.com. The homepage of this package is https://github.com/chudur-budur/nsga2-matlab. This package has included most functionality we need to use to optimizing the mixer designs or other applications, such as genetic operators, sorter, checking dominance and etc. The major part we need to customize is adding the simulation part (described above) into the main program. Please find the description of the specific functions we used from this package.

Descriptions of files in the package

- ns.ga2.m ................................................. The entrance of program
- selection.m .............................................. Selection operator
- real-cross.m ............................................ Cross operator
- real-mutate.m ........................................... Mutate operator
- mutation-pop.m ....................................... Mutate population
- fill-nondominated-sort.m ............................. Sorting operator
- evaluate-pop.m .......................... Evaluate the performance with fitness functions.
- check-dominance.m ................................. Check dominance or not