CAFES: data processing & edit guide

A. Avaro, 11/30/2021

Reference: Avaro, A.S.; Sun, Y.; Jiang, K.; Bahga, S.S.; Santiago, J.G. "Web-based open-source tool for isotachophoresis", *Analytical Chemistry*, **2021**, 93, 47, 15768–15774.

A. Export and analyze exported data from CAFES in MATLAB

- Specify and execute simulation at https://microfluidics.stanford.edu/cafes. Note the computations for these simulation run on your local computer (on your browser software).
- Use the "Save Results" button near the bottom of the web site screen to download the .zip file containing the simulation results. Extract the .zip into a "result" folder on your computer.
- Download the Python source codes from this address: https://github.com/alvinsunyixiao/itp-websim (folder "Python"). Place the "result" folder (from the simulation) in the same folder on your computer. It should be in the same folder then the "Python" folder of the source code.
- Run the python script. To do so, run a shell (on Windows, open PowerShell) and change the location to the "Python" folder. The current directory listing should end with the following: "itp-websim-master/python". Then, run the following command line:

  ```
  python matlab_converter.py -filename "../result/Simulation Results/"
  ```
  Note: If you would like to place the "result" file in another location, you should specify the corresponding directory in this command line after -filename (instead of "../result/Simulation Results/".

- The script will now prompt you to enter the name of a MATLAB table file for the analyzed data. Enter the desired name of the MATLAB table and press Enter.
- The newly created MATLAB table (.mat) file should appear within the folder where the Python script is located. This file contains the time array, the spatial grid, the electrolyte concentration profiles at all times, the hydronium concentration at all times, and the electric field profile.

You can also use the other Python script in the "Python" folder to manipulate the simulation data directly. Note that Python is completely free and open source. It is probably already installed on your computer. If not, you can download it here https://www.python.org/downloads/. Make sure to have all the modules installed (using pip: https://pip.pypa.io/en/stable/installation/).

B. How to edit the CAFES simulation tool

CAFES is free and fully open source and so you can create customized versions of the tool. We recommend this approach only for users with some experience in JavaScript and Python.

B. 1. Edit on your (local) machine

First download the source code (or better, use `git clone` to create a clone of the repository) here: https://github.com/alvinsunyixiao/itp-websim.

Install an Integrated Development Environment (IDE) to edit the JavaScript (.js) files. I strongly recommend you use the Visual Studio Code (VS Code), which allows easy edit of all the files format we use here. This tool is completely free, find it here: https://code.visualstudio.com/. Open the folder of the code in VS Code. To edit and test your code you will need Node.js and yarn. See these two tutorials to make sure Node.js, npm and yarn are installed:

- Node.js and npm: https://docs.npmjs.com/downloading-and-installing-node-js-and-npm
- Yarn: https://classic.yarnpkg.com/lang/en/docs/install/#windows-stable

Once node.js, npm and yarn are installed, run `yarn install` (in the VSCode terminal or the shell at the directory of the source code) to install all the JavaScript packages associated with CAFES which you would like to edit and test. Run `yarn start` to boot a local version of the tool on your local machine. The local version of the tool should open in your default browser. Otherwise, click on the address displayed in the terminal by VSCode. Each time you save a modification (by simply saving the file after you make

a modification), the JavaScript code will compile again automatically and display your changes. You do not need to run a command line for the changes to be updated on your local machine.

When you are finished, stop the local run on your machine by killing the process (Ctrl+C in the VSCode terminal, or simply kill the terminal).

B. 2. Edit the online version

You will need editing permission on the GitHub for this section. Contact Alvin Sun to get it. You will also need Git imperatively for this section. It is also free; you can get it here: https://git-scm.com/downloads.

- First pull the latest version of the code with `git pull`.
- Create a branch of the repository using `git checkout -b name_of_edit_branch`. (Replace name_of_your_branch by the name of your branch).
- Make your changes on the code. Make sure to save your work. As good practice, merge your changes with the master branch when you are done.
- Run `git checkout alvin/deploy`
- Run `git checkout -b name_of_deploy_branch` (you can choose the name of your deploy branch)
- Run `git merge name_of_edit_branch` (edit branch is master if you have merged the branch into master)
- Run `yarn build` (this can take a couple minutes).
- A build folder should then be created. Rename the "build" file to "cafes". In cafes/static/js and cafes/static/css, delete all the files with the extension .map. (Otherwise the deployment on the Stanford servers will not work).

The production folder is now ready. You will then need permission to get access to Stanford intranet to modify the online version. For this, ask a lab admin to run the following command line in a shell on cardinal.stanford.edu:

<div align="center">

`pts adducer -u your_SUNet_ID -g microfluidics-admins`

</div>

Now install SecureFX: https://www.vandyke.com/cgi-bin/releases.php?product=securefx. You can now have access to Stanford servers if you run this program (you will need double authentication like Duo Push). Find the group folder in ir.stanford.edu/group/microfluidics. Use the search bar on the top right of the interface.

Go to the WWW file (the complete address is afs/ir.stanford.edu/group/microfluidics/WWW). Here, replace the "cafes" folder by the one you have just created (the production folder you have prepared). Make sure to replace all files and choose binary (non-text) file transfer. Now, open your web browser at https://microfluidics.stanford.edu/cafes and make sure that your latest production build works.

C. Contact information

Juan G. Santiago: juan.santiago@stanford.edu

Alexandre S. Avaro: aavaro@stanford.edu

Yixiao "Alvin" Sun: alvinsun@stanford.edu

Kaiying "Katherine" Jiang: kaiying@stanford.edu

Supreet S. Bahga: bahga@mech.iitd.ac.in