Artifact for Efficient Trace Generation for Rare-Event Analysis in Chemical Reaction Networks

By Bryant Israelsen, Landon Taylor, and Zhen Zhang at Utah State University.

Abstract

This artifact contains the models and tools described in the paper "Efficient Trace Generation for Rare-Event Analysis in Chemical Reaction Networks" along with instructions to reproduce results similar to those in the paper.

Virtual Machine

The artifact is a virtual machine which can be run as follows:

1. Download the `ragtimer_artifact.zip` file from `https://doi.org/10.5281/zenodo.7105424` (the Artifact is hosted on Zenodo with DOI 10.5281/zenodo.7105424).
2. Verify the SHA256 checksum of the archive file is `6555ba9ff3794a8cfc6aa9ad832238cd163fd080018bd9f54cab147ddd491e5a`. Extract the `.vdi` file.
3. Load virtual machine into preferred VM client. The virtual machine should be given at least 8 GB of RAM and 12 GB of disk memory. This VM has been tested in KVM on Debian 11 and Virtualbox on Mac OS and Windows 10.
4. Log in as user `Ragtimer User` with password `ragtimer`
5. Open a terminal. Source code for RAGTIMER is found in the `~/ragtimer` directory.

Running a Ragtimer Test

This test may take approximately 30 minutes to run, depending on system specifications.

1. Navigate to the `~/ragtimer/scripts` directory using `cd ~/ragtimer/scripts`. In this folder is a shell script `artifact.sh` which will generate 10,000 traces for the three models mentioned in "Efficient Trace Generation for Rare-Event Analysis in Chemical Reaction Networks". These models are informally described below.
2. Run the shell script via `./artifact.sh`. Results will print directly on the console. If permissions are denied for `artifact.sh`, run `sudo chmod 777 artifact.sh` with sudo password `ragtimer`. Expect results to take several seconds to generate due to the nature of the virtual machine.

3. Expect a printout for each model, with the final line in each printout providing the target's probability. This is indicated by `Total Sum of Unique Path Probabilities`. Because RAGTIMER relies on randomized testing, each execution's probability will be slightly different.
4. If desired, modify the second line of `~/ragtimer/model.ragtimer` to produce an unreachable target state. Using tabs to separate, change `50 2 0 50 0 0 0` to `50 2 0 49 0 0 0`. Because the target state becomes unreachable with insufficient G, Ragtimer will report unreachability.

Note that in the shell script, we use the `loose` command-line argument. This enables loose mode, in which some non-essential reactions are enabled, allowing for the production of a wider array of paths.

Model Descriptions

The following models are tested in this artifact and described in Sections 10.1, 10.2, and 10.3 of "Efficient Trace Generation for Rare-Event Analysis in Chemical Reaction Networks".

1. **Single Species Production-Degredation Model**. In this model, the target involves the production of a single species in two reactions.
2. **Enzymatic Futile Cycle Model**. In this model, the target involves the oscillation between two of six reactions.
3. **Yeast Polarization Model**. In this model, a rare combination of three of eight reactions must fire to reach the target.