1. Maximum Caliber
   * Required files: COLVAR trajectory with header and time column removed.
   * Relevant parameters: sgoop.in\_file, sgoop.rc\_bin, sgoop.wells
   * Example file: sgoop.py
   * Instructions: Perform unbiased molecular dynamics printing the trajectory of order parameters using PLUMED. This trajectory will be stored in the COLVAR file. Trim the header and time column from the COLVAR file and use this trimmed trajectory as the sgoop.in\_file. This will then be used to calculate how often transitions between states occur using the maximum caliber framework.
2. Enhanced sampling (optional)
   * Required files: COLVAR trajectory with bias column. FES files output from the PLUMED function sumhills.
   * Relevant parameters: reweight.fesfilename, numdat, reweight.datafile, reweight.col\_rewt, reweight.col\_bias, ngrid, sparse (can help SGOOP runtime)
   * Example file: reweight.py
   * Instructions: Use an enhanced sampling technique such as metadynamics to obtain a biased trajectory of order parameters. The trajectory of these parameters will be stored in the COLVAR file as before. The SUMHILLS function in PLUMED may be used with the HILLS output file to create FES files containing free energy. Use a reweighting script such as the provided reweight.py to calculate the unbiased probability distribution along each trial reaction coordinate. The reweight.py script will use these free energies as well as the COLVAR trajectory to create an unbiased probability distribution along a given reaction coordinate.
3. Simulated Annealing
   * Relevant parameters: x0, T, stepsize (all for basinhopping)
   * Example files: Reweighting Loop.ipynb, Unbiased Loop.ipynb
   * Instructions: Perform simulated annealing or a similar optimization procedure with inputs of reaction coordinate weights and outputs of spectral gaps. For enhanced sampling this will use reweighting on the trial reaction coordinate and sgoop.sgoop on the reweighted probability. For unbiased dynamics this will use sgoop.rc\_eval on the trial reaction coordinate. This will return the optimal reaction coordinate for a given number of wells (starting at 2). Increase sgoop.wells and repeat optimization for the next number of wells and check if the returned free energy profile is self-consistent by counting the number of wells (we consider wells that are separated by barriers > kT). Repeat the process of increasing sgoop.wells and optimizing the spectral gap until an inconsistent result is returned. The result corresponding to n-1 wells where n is the first inconsistent result is the overall optimal reaction coordinate.
   * Note that in the provided examples run times are printed. If these are too high, the stride of input files may be changed to reduce run time.
   * The default sparse=False argument in the reweighting methods will trim off negligible probabilities values from the ends of the returned probability array. This argument is beneficial for SGOOP as it has trouble with sparse arrays. Sparse=True should be used when the probability is not being used as a SGOOP input (ie input for rebiasing or counting wells). Sparse=True may also be beneficial for systems that have important features with very low probability as these would be neglected otherwise.
4. Iterative Enhanced Sampling (optional)
   * Relevant parameters: see Enhanced Sampling and Simulated Annealing
   * Instructions: The enhanced sampling and simulated annealing steps can be repeated using the SGOOP reaction coordinate to add bias. Change the metadynamics reaction coordinate in PLUMED to match the SGOOP reaction coordinate. Repeat the simulated annealing process with the new COLVAR and FES files from subsequent metadynamics runs while using the same unbiased COLVAR file for maximum caliber calculations. This will improve sampling allow for more accurate SGOOP calculations. After a few iterations of SGOOP and molecular dynamics, the reaction coordinate should converge.
5. Conditional Probability Factorization (optional)
   * Relevant parameters: see Enhanced Sampling and Simulated Annealing
   * Instructions: If one reaction coordinate is not sufficient for sampling (ie some features are hidden) more reaction coordinates may be generated. Use the function rebias instead of the function reweight to generate a probability conditional on the first reaction coordinate. A trajectory with bias along the first reaction coordinate should be used for maximum caliber as this trajectory will have a free energy surface that is close to that of the conditional probability.
   * Note that the old probability used as an input for the conditional probability should be generated using the sparse=True argument in order to account for the full probability along the first reaction coordinate.