README Documentation

Universal kinetic solvent effects in acid-catalyzed reactions of biomass-derived oxygenates

Theodore W. Walker[a], Alex K. Chew[a], Huixiang Li[bc], Benginur Demir[ad], Z. Conrad Zhang[b], George W. Huber[a], Reid C. Van Lehn[a†], and James A. Dumesic[ad]

[†]Please send all e-mail correspondence to: vanlehn@wisc.edu
[a]Department of Chemical and Biological Engineering, University of Wisconsin-Madison, Madison, USA.
[b]Dalian National Laboratory for Clean Energy, Dalian Institute of Chemical Physics, Chinese Academy of Sciences, 457 Zhongshan Road, Dalian 116023, China
[c]University of Chinese Academy of Sciences, Beijing 100049, China
[d]DOE Great Lakes Bioenergy Research Center, University of Wisconsin-Madison, Madison, USA

September 11, 2019

Contents

1 Description

The purpose of this document is to go through a step-by-step procedure in developing simulation data for the article:

Walker, T. W.*; Chew, A. K.*; Li, H.; Demir, B.; Zhang, Z. C.; Huber, G.; Van Lehn, R. C.; and, Dumesic. J. “University kinetic solvent effects in acid-catalyzed reactions of biomass-derived oxygenates” *Energy & Environmental Science*, **2018**, *11*, 617-628. [Link]

Please cite this paper if you use the codes generated here.

2 Classical molecular dynamics simulations

2.1 General idea

All classical molecular dynamics (MD) simulations were performed with Gromacs 2016. This section goes through how the classical simulations were set up and performed. The general simulation setup is shown in Figure 1. First, water and cosolvent (if applicable) is added to a (6 nm)^3 cubic box system at a desired mass fraction. After NPT equilibration, one reactant molecule is added to the system. Subsequent NPT productions were performed to compute three molecular descriptors: preferential exclusion coefficient (γ), hydrogen bonding lifetime (τ), and accessible hydroxyl fraction (δ). Please refer to the main text for additional details of these molecular descriptors.

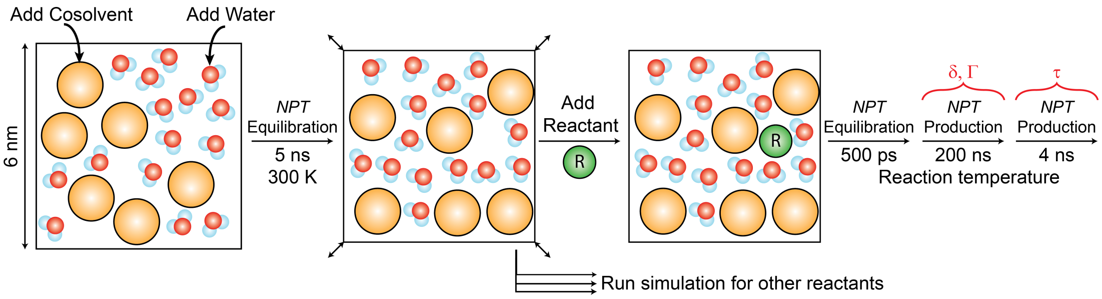


Figure 1: General workflow of generating mixed-solvent environments. “R” denotes the reactant. Note that the second production trajectory, used to calculate hydrogen bonding lifetimes, was not always 4 ns; some reactants required a longer simulation time to obtain accurate hydrogen-bonding lifetime data.

2.2 Data availability

All simulation data is freely available in Zenodo.
To access the data, click on the link, download the file, and unzip the directory. The entire directory is approximately 72 GB. The command to unzip in Linux terminal is:

```
tar -zxvf NAME.tar.gz
```

where “NAME” is the name of the zipped file.

2.3 Software requirements

All software requirements are listed:

- GROMACS 2016 (Version 0) - Used to run MD simulations
- GROMACS 5.0.1 - Used to compute hydrogen bonding lifetimes
- Python 2.7 - Used to generate force field parameters for molecules
- Python 3.4 - Used to compute molecular descriptors

2.4 Directory structure

Upon downloading the Zenodo link, the directory contains the following files:

- `analysis_scripts`: Directory with analysis scripts to compute molecular descriptors. (Section 3)
- `BuildingSystem`: Directory to generate cosolvent and reactant force field parameters (Section 2.5)
- `excel_spreadsheet`: Directory that contains spreadsheet data of simulations and experiments. (Section 4)
- `prep_mixed_solvents`: Simulations of mixed-solvent environments in various mass fractions (Section 2.6)
- `Scripts`: Contains all scripts required to run simulations.
- `Simulations`: Contains all simulation data. (Section 2.8)

2.5 Generating cosolvent and reactant forcefield parameters

Forcefield parameters were generated with the CGenFF/CHARMM36 all-atom forcefield. The procedure for generating reactant and cosolvent molecule is listed below:

- Reactant or cosolvent molecular structures were found in https://zinc.docking.org/, an extensive database with accurate structural information.
- Download the *.mol2 file from the ZINC website.
- Generate a *.str file from the CGenFF.paramchem.org website. Note that you may need an acccount to add a molecule. On the website, click “Upload Molecule”, choose the *.mol2 file, then select “Use CGenFF legacy v1.0”. Save the *.str and *.mol2 file to the `BuildingSystem` directory.
- To prepare the system, first source the bashrc: `source MAIN_dir/Scripts/bin/bash_rc.sh` where `MAIN_dir` is the main directory of copied from Zenodo.

- Go to the `BuildingSystem` directory. For this example, we use xylitol reactant as an example. Run the command: `prep_charmm36_molecule xylitol XYL`, where "XYL" is the residue name. Listing 1 shows the output of the command. Now, you should have a directory `xylitol`, which has the *.itp, *.pdb, *.prm, and *.top file necessary to run a simulation.

```
*** RUNNING PREPARATION CHARMM36 ***
MOLECULE FULL NAME: xylitol
MOLECULE RESIDUE NAME: XYL
FIXING RESIDUE NAMES OF MOL2 AND STR TO: XYL
RUNNING COMMAND: ./cgenff_charmm2gmx.py XYL xylitol.mol2 xylitol.str charmm36-nov2016.ff
NOTE1: Code tested with python 2.7.3. Your version: 2.7.5 (default, Aug  4 2017, 00:39:18)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)]

NOTE2: Please be sure to use the same version of CGenFF in your simulations that was used during parameter generation:
--Version of CGenFF detected in  xylitol.str : 3.0.1
--Version of CGenFF detected in  charmm36-nov2016.ff/forcefield.doc : 3.0.1

NOTE3: In order to avoid duplicated parameters, do NOT select the 'Include parameters that are already in CGenFF' option when uploading a molecule into CGenFF.
============ DONE ============
Conversion complete.
The molecule topology has been written to xyl.itp
Additional parameters needed by the molecule are written to xyl.prm, which needs to be included in the system .top
============ DONE ============
CREATING DIRECTORY: xylitol
COMPLETION -- MOVED ALL PREPARATION FILES TO: xylitol
```

Listing 1: Output of `prep_charmm36_molecule`

- After generating the force field parameters, the files are moved to a more convenient place and used to prepare for mixed-solvent simulations.
  - If you have reactants, run the command: `solvent_effects_add_molecule xylitol solutes` This command simply copies the *.itp, *.prm, and *.pdb of xylitol to:
    `MAIN_dir/prep_mixed_solvent/input_files/solutes/xylitol`
    This will also create a *.gro file using `gmx editconf` command.
  - If you are adding a solvent, you will need to include the residue name, *e.g.* dioxane:
    `solvent_effects_add_molecule dioxane solvents DIO`
    This will run a 500 ps NPT equilibration with 125 molecules of your solvent at $T = 298.15$ K and $P = 1$ bar.

Now, you should have all reactant and cosolvent molecules ready for mixed-solvent environment simulations. The directories for solute and solvent are shown in Listing 2 and 3.

```
12-propanediol
cellobiose
ETBE
fructose
levoglucosan
```

```
tBuOH
xylitol
```

Listing 2: Directories within MAIN_dir/prep_mixed_solvent/input_files/solutes

```
dioxane
GVL_L
molecular_volume.info
molecular_weights.info
molecular_weights_ref.txt
prep_solvents
spc216
tetrahydrofuran
```

Listing 3: Directories within MAIN_dir/prep_mixed_solvent/input_files/solvents

2.6 Generating mixed-solvent environments MD simulations

For mixed-solvent systems, reactants and cosolvents are initially added to a 6 nm box, then simulated for 5 ns. The procedure to run mixed-solvent system is described below:

- Go to directory: `MAIN_dir/Scripts/Prep_Scripts`
- Modify the code: `full_prep_cosolvent.sh`. You will need to modify the script to select specific reactant and cosolvents.

```
## DEFINING COSOLVENTS
declare -a cosolvent_name=("dioxane" "GVL_L" "tetrahydrofuran")
## COSOLVENT RESNAME
declare -a cosolvent_resname=("DIO" "GVLL" "THF")

## Weight fractions
declare -a solvent_one_wt_frac=("0.10" "0.25" "0.50" "0.75")

## Desired box lengths
# "4" "6" "8" "10" "12"
declare -a desire_box_lengths=("6")

## Declaring water model
water_model="spce" # tip3p, spce
```

Listing 4: Code to edit within full_prep_cosolvent.sh

- Adjust the `cosolvent_name`, `cosolvent_resname`, `solvent_one_wt_frac`, and `desire_box_lengths` variables to the desired cosolvent mixture. Note that `solvent_one_wt_frac` is the weight fraction of water. So when `solvent_one_wt_frac=10`, that means you have 10 wt% water and 90 wt% cosolvent.
- Run the bash script: `bash full_prep_cosolvent.sh` This will create mixed solvents within: `MAIN_dir/prep_mixed_solvents/prep_mixed_cosolvents/spce/6nm`

  Within this directory, you should have the directories in Listing 5.

```
100_Water
10_Water_DIO
10_Water_GVLL
10_Water_THF
25_Water_DIO
25_Water_GVLL
25_Water_THF
50_Water_DIO
50_Water_GVLL
50_Water_THF
75_Water_DIO
75_Water_GVLL
75_Water_THF
```

Listing 5: Directories within `6nm` folder

The `100_water` folder contains pure water placed in a (6 nm)^3 simulation box. Now, you should be able to access and generate all mixed-solvent environment simulations. Note: sometimes `gmx solvate` fails to correctly solvate the system, resulting in too many molecules within the *.top file that do not correspond with the *.gro file. If this is the case, then you may need to re-run the `prep_charmm36_molecule` command, but change the `want_long_equil` to `True` within:

`MAIN_dir/Scripts/Prep_Scripts/prep_solvent.sh`

This would enable a longer simulation (1.5 ns instead of 500 ps) for correct solvation of cosolvents. Alternative to this approach is to use some other software to solvate the system (*e.g.* PACKMOL), which is outside the scope of this text.

2.7 Generating reactant in mixed-solvent environment MD simulations

Once the mixed-solvent environment is developed, we then added one reactant and performed a *NPT* simulation for 200 ns. The procedure to do this is shown below:

- Open the file: `MAIN_dir/Scripts/Full_System_Prep_cosolvents.sh`
- Edit the following variables:

```
## SOLUTE NAMES
declare -a solute_names=("ETBE" "tBuOH" "levoglucosan" "12-propanediol" "
    fructose" "cellobiose" "xylitol")

## DEFINING RESIDUE NAMES
declare -a solute_residue_names=("ETBE" "tBuOH" "LGA" "PDO" "FRU" "CEL" "XYL")

## DEFINING TEMPERATURES
declare -a solute_temps=("343.15" "363.15" "403.15" "433.15" "373.15" "403.15"
    "403.15")

## WATER MASS FRACTIONS
declare -a solvent_one_wt_fractions=("0.10" "0.25" "0.50" "0.75" "1.00")

## DEFINING COSOLVENT NAME
declare -a solvent_two_molecules_names=("dioxane" "GVL_L" "THF")

## DEFINING COSOLVENT RESIDUE NAME
```

```
declare -a solvent_two_molecules_residue_names=("DIO" "GVLL" "THF")

## Desired box lengths
# "4" "6" "8"
declare -a desire_box_lengths=("6")
```

Listing 6: Code to edit within `Full_System_Prep_cosolvents.sh`

Listing 6 generates simulation for all 7 reactant systems for aqueous mixtures of dioxane (DIO), γ-valerolactone (GVL), or tetrahydrofuran (THF). Note that each simulation is 200 ns long, so it will take some time to perform them. All completed reactant in mixed-solvent simulations are discussed in Section 2.8

- After these simulations are complete, an additional 5 ns production simulation is performed to compute hydrogen bonding lifetimes. To do this, change the `one_system_script_name` variable shown in Listing 7.

```
### TYPE OF SIMULATION
#one_system_script_name="One_System_Prep_cosolvents.sh"
one_system_script_name="One_System_Prep_prod_extension.sh"
```

Listing 7: Additional production simulations for hydrogen bonding lifetime in `Full_System_Prep_cosolvents.sh`

The `One_System_Prep_prod_extension.sh` takes the completed reactant-solvent production simulations and generates an additional 5 ns production simulation with more frequent outputs.

2.8 Description of available simulation data

All available simulation data is shown in: `MAIN_dir/Simulations/main_simulation`
Listing 8 shows the directories in `main_simulation`.

```
CEL
ETBE
FRU
LGA
PDO
tBuOH
XYL
```

Listing 8: Directories within `MAIN_dir/Simulations/main_simulation`

Each directory in Listing 8 corresponds to the residue name. For example, “XYL” represents xylitol. Listing 9 shows the directories within XYL.

```
mdRun_403.15_6_nm_XYL_100_WtPercWater_spce_Pure
mdRun_403.15_6_nm_XYL_10_WtPercWater_spce_dioxane
mdRun_403.15_6_nm_XYL_10_WtPercWater_spce_GVL_L
mdRun_403.15_6_nm_XYL_10_WtPercWater_spce_tetrahydrofuran
mdRun_403.15_6_nm_XYL_25_WtPercWater_spce_dioxane
mdRun_403.15_6_nm_XYL_25_WtPercWater_spce_GVL_L
mdRun_403.15_6_nm_XYL_25_WtPercWater_spce_tetrahydrofuran
mdRun_403.15_6_nm_XYL_50_WtPercWater_spce_dioxane
mdRun_403.15_6_nm_XYL_50_WtPercWater_spce_GVL_L
mdRun_403.15_6_nm_XYL_50_WtPercWater_spce_tetrahydrofuran
```

```
mdRun_403.15_6_nm_XYL_75_WtPercWater_spce_dioxane
mdRun_403.15_6_nm_XYL_75_WtPercWater_spce_GVL_L
mdRun_403.15_6_nm_XYL_75_WtPercWater_spce_tetrahydrofuran
```

Listing 9: Directories within `MAIN_dir/Simulations/main_simulation/XYL`

The directory name was designed to inform about the simulation parameters.
For example, `mdRun_403.15_6_nm_XYL_10_WtPercWater_spce_dioxane` means XYL was simulated with 10 wt% water/90 wt% dioxane at 403.15 K with the initial box length of 6 nm. Each directory contains sufficient information to perform additional simulations. Here is a description of the main files:

- `mixed_solv_equil.xtc`: *NPT* equilibration simulation performed for 500 ps.
- `mixed_solv_prod.gro`: Structure file after 200 ns *NPT* production.
- `mixed_solv_prod.xtc`: *NPT* production simulation containing 200 ns worth of production data.

Each directory typically contains a `prod_extend` directory that contains additional production simulations used to compute hydrogen bonding lifetimes. Since these directories are large to store on Zenodo, they are omitted from the available simulations list. The code used to generate these simulations are available, described in Section 2.7. These simulations are available upon request.

3 Analysis of molecular dynamics simulations

All analysis scripts used to generate the molecular descriptors is available in:
`MAIN_dir/analysis_scripts`
Note that these tools could be improved, which is a subject of future work. This section outlines how the three descriptors (Γ, τ, and δ) were computed.

3.1 Extracting molecular dynamics simulations

Prior to analysis, MD simulations were extracted by taking the last 190 ns of production simulation. To make analysis easier, all important production simulation files (*e.g.* *prod.xtc) were copied to a separate directory called "Analysis". After copying simulation information, the production trajectory is extracted using `gmx trjconv` commands. Given a complete simulation, you can run the procedure below to extract MD simulations:

- Open bash script:

  `MAIN_dir/Scripts/analysis_full_extraction.sh`
- Adjust the following variables shown in Listing 10.

```
## DEFINING INPUT DIRECTORY (SIMULATION FOLDER WITH MANY SIMULATIONS)
input_dir_name="INPUT_DIRECTORY"

## DEFINING OUTPUT DIRECTORY
output_dir_name="OUTPUT_DIRECTORY"

## PROD EXTEND
```

```
alternative_extraction="prod_extend" # Turn on if you want prod extend
alternative_extraction=""
```

Listing 10: Variables in `analysis_full_extraction.sh`

- Change the `input_dir_name` to the directory within the `Simulations` folder. Note that this directory should have all simulations listed (*e.g.* `mdRun_403.15_6_nm_XYL_10_WtPercWater_spce_dioxane`, and so on). Adjust the `output_dir_name` to a desired output directory name. If you want to extract the `prod_extend` simulations, alter the `alternative_extraction` variable to `prod_extend`

- Run the bash script: `bash analysis_full_extraction.sh`

- Now, you should have an `Analysis` folder which contains `output_dir_name` directory. For the analysis, we combined directories with the same residue name (*e.g.* tBuOH, XYL). To do this, go into the `output_dir_name` directory and run in the terminal `moveSimilar tBuOH`. Repeat for all residue names (*e.g.* LGA, XYL, *etc.*). This will create a directory and move all tBuOH directories within it. Note that the `moveSimilar` command can be accessed when sourcing the bashrc: `source MAIN_dir/Scripts/bin/bash_rc.sh`

3.2 Computing prefential exclusion coefficients (Γ)

Preferential interaction coefficients were computed in Python 3.4 using the *MDTraj* module. The procedure is:

- Open python file within `analysis_scripts`:

  `analysisTools_preferential_interactions/analysisTools_preferential_interactions.py`

- Change the variables shown in Listing 11.

```
## Defining directories for analysis ##
directory4Analysis =r"180323-12-PDO_SIMS"

# Defining directories for analysis
path2AnalysisDir=r"R:\scratch\SideProjectHuber\Analysis\\" + directory4Analysis # PC Side

# Checking if we are on the server (SWARM)
path2AnalysisDir = checkPath2Server(path2AnalysisDir)

# Defining directory structure
# 'tBuOH', 'XYL', 'FRU', 'ETBE', 'PDO', 'CEL', 'LGA'
analysisDir = ['PDO', 'tBuOH']
## DEFINING STRUCTURE FILES
structureFile="mixed_solv_prod.gro"
xtcFile= 'mixed_solv_prod_10_ns_whole.xtc'

# Defining residues we are interested in
solute_resname = analysisDir[:]
cosolvent_resnames = ["GVLL", "DIO", "THF"]
water_resname = "HOH" # Does not change
```

Listing 11: Variables in `analysisTools_preferential_interactions.py`

- Change variables to match the output simulations from Section 3.1. The description of the variables is shown:
  - `directory4Analysis`: directory for analysis.
  - `path2AnalysisDir`: path to your analysis directory.
  - `analysisDir`: analysis directory names, described at the end of Section 3.1.
  - `structureFile`: gro or pdb file from production simulation.
  - `xtcFile`: xtc file name from production simulation.
  - `cosolvent_resnames`: list of cosolvent residue names to look for.
- Run the script: `python3 analysisTools_preferential_interactions.py` Note that we use `python3` as an alias for Python 3.4.
- After running the script, you will need to extract the preferential exclusion coefficient from a generated pickle file. Pickle files is a pythonic way to store variables. Listing 12 shows the codes that need to be adjusted for extracting preferential exclusion coefficients.

```
### RETRIEVING VARIABLES ###

import sys

if sys.prefix == 'C:\\Users\\akchew\\AppData\\Local\\Continuum\\Anaconda3\\envs\\py27' or sys.prefix == '/Users/alex/anaconda/envs/py27' or sys.prefix == "C:\\Users\\akchew\\AppData\\Local\\Continuum\\Anaconda3": # on PC/MAC

# Getting back objects
import pickle

# Variable directories
MainAnalysisDir=r"R:\scratch\SideProjectHuber\Scripts\AnalysisScripts\analysisTools_preferential_interactions\pickle_analysisTools_preferential_interactions"
## DEFINING PICKLE FILE
pickleFile=r"180323-analysisTools_preferential_interactions_variables.pickle"
```

Listing 12: Extraction of `analysisTools_preferential_interactions.py`

Here, you will need to change `pickleFile` variable to the pickle that was outputted. Run the retrieval protocol to extract preferential exclusion coefficients.

3.3 Computing hydrogen bonding lifetimes (τ)

Hydrogen bonding lifetimes were generated using the `prod_extend` directories, extracted using the protocol described in Section 3.1. The procedure to extract hydrogen bonding lifetimes is shown below:

- We have found that GROMACS 2016 incorrectly computes hydrogen bonding lifetimes. Therefore, we compute hydrogen bonding lifetimes with an older version: GROMACS 5.0.1. An example script to download this version is shown in the following script:

  `MAIN_dir/Scripts/bin/gromacs_install.sh`

  You can download gromacs by running: `bash gromacs_install.sh`

  This will install GROMACS 5.0.1 within your home directory.

- Once GROMACS 5.0.1 is installed, you will need to load it. Below are functions in terminal that could be stored in your /home/$USER/.bashrc to load GROMACS 5.0.1:

```bash
## GENERAL FUNCTION TO LOAD GROMACS
function general_gromacs_load () {
  # $1: gromacs version (gromacs_2016)
  # $2: type: thread or mpi
  echo "Loading $1 Version with $2 selection"
  current_MPI_Selection="$2"
  if [[ ${current_MPI_Selection} == "thread" ]]; then
  source "${gromacs_installation_folder}/$1/${gromacs_thread_bin}/GMXRC"; PATH="$PATH:${gromacs_thread_bin}"
  elif [[ ${current_MPI_Selection} == "mpi" ]]; then
  source "${gromacs_installation_folder}/$1/${gromacs_mpi_bin}/GMXRC"; PATH="$PATH:${gromacs_mpi_bin}"
  else
  echo "Error, no thread/mpi selected! Exiting"; exit
  fi
}
## FUNCTION TO LOAD GROMACS 5.0.1
function load_gromacs_5_0_1_thread () {
  general_gromacs_load gromacs-5.0.1 thread
}
```

Listing 13: Functions to load GROMACS 5.0.1

- Load GROMACS 5.0.1 by running: load_gromacs_5_0_1_thread

- Since we are using a different GROMACS version, all *.tpr files need to be reloaded with this version. To do this, open: MAIN_dir/Scripts/Full_System_Prep_cosolvents.sh

  - Change the following variables:

    ```bash
    one_system_script_name="One_System_post_sim.sh"
    ## DEFINING ANALYSIS DIRECTORY
    main_analysis_dir="180521-20_HMF_rerun_10ns_prod_extend"
    ```

Listing 14: Variables to reload *.tpr files in Full_System_Prep_cosolvents.sh

    main_analysis_dir is the main analysis directory outputted from Section 3.1.

  - Run the code: bash Full_System_Prep_cosolvents.sh
  - Check to see if *.tpr files have been loaded into the analysis directory

- Now that you have all *.tpr files reloaded with the correct GROMACS version, open bash file within analysis_scripts: analysisTools_HBond.sh

- Listing 15 shows the variables to adjust for computing hydrogen bonding lifetimes.

```bash
## DEFINING LOGICALS
skipGMX="False" # True if you want to skip the gmx hbond (may produce errors if you haven't run this script already)
wantSummary="True" # True if you want to extract all the data
wantIndex="False" # True if you want index groups
wantGeneralOH="True" # True if you want the analysis of the entire OH as well! -- Does not overwrite "wantOHGroups"
wantOHGroups="False" # True if you want analysis of each individual hydroxyl group
```

```

## DEFINING FIRST FRAME TO START READING
first_frame="1000" # ps

## SPLIT HBONDING ##
wantSplitOH="True" # True if you want splitting hydrogen bonds
split_time="2000"
total_sim_time="5000"

## Summary characteristics ##
wantHBondvsTime="True" # True if you want hydrogen bonds over time for each file

# Loading GROMACS #
load_gromacs_5_0_1_thread

## DEFINING ANALYSIS DIRECTORY
specific_analysis_dir="170826-7Molecules_200ns_Combed_WithITP_3Solvents_prod_extend" # Directory where you want specifically the analysis to occur

# Extraction directories within your folder
# "CEL" "ETBE" "PDO" "FRU" "tBuOH" "LGA" "XYL"
declare -a extractDirList=("tBuOH")
```

Listing 15: Variables in `analysisTools_HBond.sh`

The description of important variables is shown below:

- `skipGMX`: True if you want to skip gmx hbond (assuming you already ran it)
- `wantSummary`: True if you want a summary file that has all hydrogen bonding lifetime information
- `first_frame`: First frame to start analysis. Currently set at 1 ns.
- `wantSplitOH`: True if you want to split the trajectory to run hydrogen bonding analysis.
- `split_time`: Time in ps to truncate trajectory, set at 2,000 ps (2 ns).
- `total_sim_time`: Total time in ps used in the simulation, set at 5,000 ps (5 ns).
- `extractDirList`: List of residue names within your analysis folder.

You can run the code by: `bash analysisTools_HBond.sh`

Note that this may take some time to run depending on your available computational resources.

3.4 Computing accessible hydroxyl fractions (δ)

Accessible hydroxyl fractions are computed with the procedure below:

- Open the python within `analysis_scripts`:

  `analysisTools_SASA/analysisTools_SASA.py`

- Edit the variables shown in Listing 16.

```
## Defining directories for analysis ##
directory4Analysis =r"180323-12-PDO_SIMS"

# Defining directories for analysis
path2AnalysisDir=r"R:\scratch\SideProjectHuber\Analysis\\" + directory4Analysis
     # PC Side

# Checking if we are on the server (SWARM)
path2AnalysisDir = checkPath2Server(path2AnalysisDir)

# Defining directory structure
analysisDir = ['tBuOH']
## FINDING GRO AND XTC FILE
xtcFile = "mixed_solv_prod_10_ns_whole.xtc"
groFile = 'mixed_solv_prod_structure.pdb'

# Defining itp files
itp_file_names = ['tBuOH.itp']

## SASA Variables ##
Probe_radius = 0.14 # in nms
Num_Sphere_Pts = 960 # Number of sphere points

# Defining SASA type
current_SASA_type = 'alcohol2allSASA'
wantCustomSASA = True
# True if you want VDW from Bondi

# Defining full directory path
fullpath2Dir = [ path2AnalysisDir +'/'+ x for x in analysisDir]

# Defining residue names
residueNames= ['tBuOH']
```

Listing 16: Variables within `analysisTools_SASA/analysisTools_SASA.py`

The variables are defined below:

- `directory4Analysis`: analysis directory for analysis.
- `path2AnalysisDir`: path to analysis directory.
- `analysisDir`: list of analysis directories
- `xtcFile`: production *.xtc file.
- `groFile`: production *.gro file.
- `itp_file_names`: *.itp file names within analysis directory
- `Probe_radius`: probe radius used for computing solvent-accessible-surface area (SASA)
- `Num_Sphere_Pts`: number of sphere points used for SASA
- `current_SASA_type`: type used to compute SASA. The `alcohol2allSASA` type was used to compute the δ descriptor.
- `wantCustomSASA`: True if you want to use Bondi Van der Waals radii.
- `residueNames`: list of residue names, which should match the itp file.

- You can run the python script: `python3 analysisTools_SASA.py`

- The script should output a pickle file with the stored SASA. You can analyze the pickle with the same script by changing the variables shown in Listing 17.

```
import sys
if sys.prefix == 'C:\\Users\\akchew\\AppData\\Local\\Continuum\\Anaconda3\\envs\\py27' or \
sys.prefix == '/Users/alex/anaconda/envs/py27' or sys.prefix == "C:\\Users\\akchew\\AppData\\Local\\Continuum\\Anaconda3": # on PC/MAC

# Getting back objects
import pickle

# Variable directories
MainAnalysisDir=r"R:\scratch\SideProjectHuber\Scripts\AnalysisScripts\analysisTools_SASA\pickle_analysisTools_SASA"
pickleFile=r"180323-analysisTools_SASA_variables.pickle"
```

Listing 17: Analysis variables within `analysisTools_SASA/analysisTools_SASA.py`

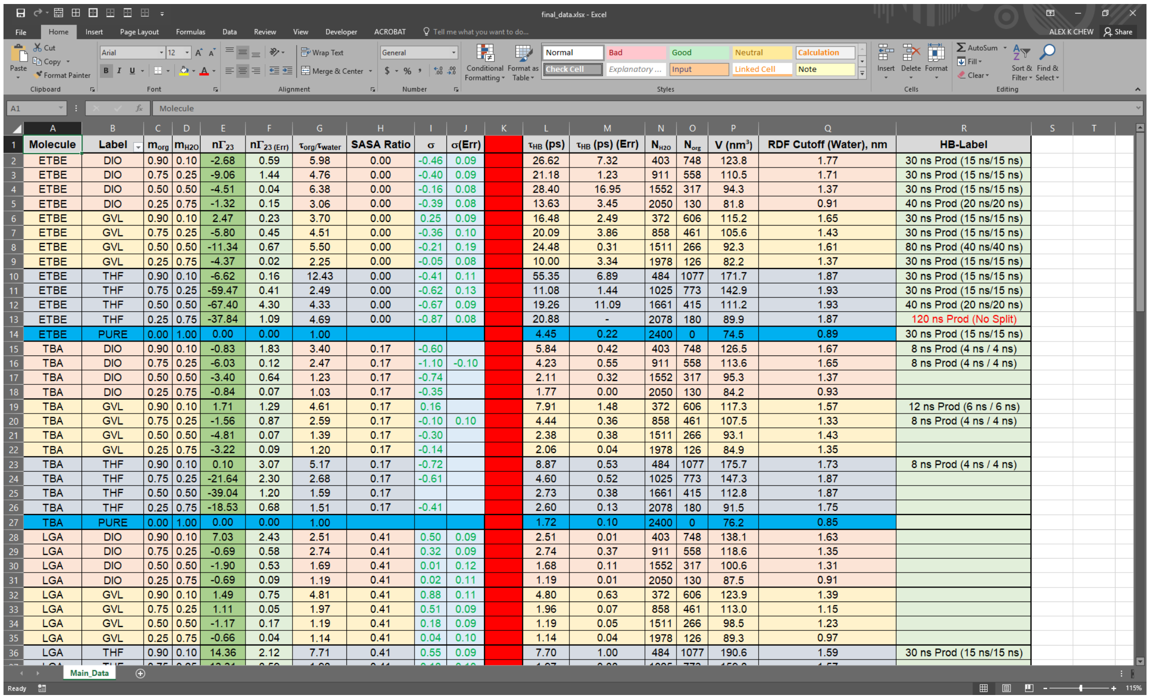
Here, you change the `MainAnalysisDir` path variable and `pickleFile` to the output name from running the `analysisTools_SASA.py` script.

4 Available tabulated data

Descriptors are tabulated in the following directory:
`MAIN_dir/excel_spreadsheet/final_data.xlsx`
Figure 2 shows a snapshot of the spreadsheet. Each row contains a simulation or experiment performed for a specified “Molecule” in a solvent composition. The variables for each column is shown below:

- **Column B**: Label for cosolvent. DIO means dioxane, GVL means γ-valerolactone, THF means tetrahydrofuran, and PURE means pure water.
- **Column C**: Mass fraction of the organic cosolvent.
- **Column D**: Mass fraction of the water.
- **Column E**: Preferential exclusion coefficient average value, computed across two 95 ns partitions.
- **Column F**: Error for the preferential exclusion coefficient.
- **Column G**: Ratio of hydrogen bonding lifetimes between the mixed-solvent environment and pure water.
- **Column H**: δ descriptor values.
- **Column I**: Experimental solvent kinetic parameters
- **Column J**: Errors in the experimental solvent kinetic parameters
- **Column L**: Hydrogen bonding lifetimes between reactant and water.
- **Column M**: Error in the hydrogen bonding lifetimes.

- **Column N**: Number of water molecules in the simulation system.
- **Column O**: Number of cosolvent molecules in the simulation system.
- **Column P**: Volume of the simulation box in nm^3.
- **Column Q**: Radial cutoff used to compute the preferential exclusion coefficient in nm.
- **Column R**: Specific systems that required a longer production simulation to compute hydrogen bonding lifetimes.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Molecule	Label	m_{org}	m_{H2O}	$n\Gamma_{23}$	$n\Gamma_{23\,(Err)}$	τ_{org}/τ_{water}	SASA Ratio	σ	σ(Err)		τ_{HB} (ps)	τ_{HB} (ps) (Err)	N_{H2O}	N_{org}	V (nm^3)	RDF Cutoff (Water), nm	HB-Label
2	ETBE	DIO	0.90	0.10	-2.68	0.59	5.98	0.00	-0.46	0.09		26.62	7.32	403	748	123.8	1.77	30 ns Prod (15 ns/15 ns)
3	ETBE	DIO	0.75	0.25	-9.06	1.44	4.76	0.00	-0.40	0.09		21.18	1.23	911	558	110.5	1.71	30 ns Prod (15 ns/15 ns)
4	ETBE	DIO	0.50	0.50	-4.51	0.04	6.38	0.00	-0.16	0.08		28.40	16.95	1552	317	94.3	1.37	30 ns Prod (15 ns/15 ns)
5	ETBE	DIO	0.25	0.75	-1.32	0.15	3.06	0.00	-0.39	0.08		13.63	3.45	2050	130	81.8	0.91	40 ns Prod (20 ns/20 ns)
6	ETBE	GVL	0.90	0.10	2.47	0.23	3.70	0.00	0.25	0.09		16.48	2.49	372	606	115.2	1.65	30 ns Prod (15 ns/15 ns)
7	ETBE	GVL	0.75	0.25	-5.80	0.45	4.51	0.00	-0.36	0.10		20.09	3.86	858	461	105.6	1.43	30 ns Prod (15 ns/15 ns)
8	ETBE	GVL	0.50	0.50	-11.34	0.67	5.50	0.00	-0.21	0.19		24.48	0.31	1511	266	92.3	1.61	80 ns Prod (40 ns/40 ns)
9	ETBE	GVL	0.25	0.75	-4.37	0.02	2.25	0.00	-0.05	0.08		10.00	3.34	1978	126	82.2	1.37	30 ns Prod (15 ns/15 ns)
10	ETBE	THF	0.90	0.10	-6.62	0.16	12.43	0.00	-0.41	0.11		55.35	6.89	484	1077	171.7	1.87	30 ns Prod (15 ns/15 ns)
11	ETBE	THF	0.75	0.25	-59.47	0.41	2.49	0.00	-0.62	0.13		11.08	1.44	1025	773	142.9	1.93	30 ns Prod (15 ns/15 ns)
12	ETBE	THF	0.50	0.50	-67.40	4.30	4.33	0.00	-0.67	0.09		19.26	11.09	1661	415	111.2	1.93	40 ns Prod (20 ns/20 ns)
13	ETBE	THF	0.25	0.75	-37.84	1.09	4.69	0.00	-0.87	0.08		20.88	-	2078	180	89.9	1.87	120 ns Prod (No Split)
14	ETBE	PURE	0.00	1.00	0.00	0.00	1.00					4.45	0.22	2400	0	74.5	0.89	30 ns Prod (15 ns/15 ns)
15	TBA	DIO	0.90	0.10	-0.83	1.83	3.40	0.17	-0.60			5.84	0.42	403	748	126.5	1.67	8 ns Prod (4 ns / 4 ns)
16	TBA	DIO	0.75	0.25	-6.03	0.12	2.47	0.17	-1.10	-0.10		4.23	0.55	911	558	113.6	1.65	8 ns Prod (4 ns / 4 ns)
17	TBA	DIO	0.50	0.50	-3.40	0.64	1.23	0.17	-0.74			2.11	0.32	1552	317	95.3	1.37	
18	TBA	DIO	0.25	0.75	-0.84	0.07	1.03	0.17	-0.35			1.77	0.00	2050	130	84.2	0.93	
19	TBA	GVL	0.90	0.10	1.71	1.29	4.61	0.17	0.16			7.91	1.48	372	606	117.3	1.57	12 ns Prod (6 ns / 6 ns)
20	TBA	GVL	0.75	0.25	-1.56	0.87	2.59	0.17	-0.10	0.10		4.44	0.36	858	461	107.5	1.33	8 ns Prod (4 ns / 4 ns)
21	TBA	GVL	0.50	0.50	-4.81	0.07	1.39	0.17	-0.30			2.38	0.38	1511	266	93.1	1.43	
22	TBA	GVL	0.25	0.75	-3.22	0.09	1.20	0.17	-0.14			2.06	0.04	1978	126	84.9	1.35	
23	TBA	THF	0.90	0.10	0.10	3.07	5.17	0.17	-0.72			8.87	0.53	484	1077	175.7	1.73	8 ns Prod (4 ns / 4 ns)
24	TBA	THF	0.75	0.25	-21.64	2.30	2.68	0.17	-0.61			4.60	0.52	1025	773	147.3	1.87	
25	TBA	THF	0.50	0.50	-39.04	1.20	1.59	0.17				2.73	0.38	1661	415	112.8	1.87	
26	TBA	THF	0.25	0.75	-18.53	0.68	1.51	0.17	-0.41			2.60	0.13	2078	180	91.5	1.75	
27	TBA	PURE	0.00	1.00	0.00	0.00	1.00					1.72	0.10	2400	0	76.2	0.85	
28	LGA	DIO	0.90	0.10	7.03	2.43	2.51	0.41	0.50	0.09		2.51	0.01	403	748	138.1	1.63	
29	LGA	DIO	0.75	0.25	-0.69	0.58	2.74	0.41	0.32	0.09		2.74	0.37	911	558	118.6	1.35	
30	LGA	DIO	0.50	0.50	-1.90	0.53	1.69	0.41	0.01	0.12		1.68	0.11	1552	317	100.6	1.31	
31	LGA	DIO	0.25	0.75	-0.69	0.09	1.19	0.41	0.02	0.11		1.19	0.01	2050	130	87.5	0.91	
32	LGA	GVL	0.90	0.10	1.49	0.75	4.81	0.41	0.88	0.11		4.80	0.63	372	606	123.9	1.39	
33	LGA	GVL	0.75	0.25	1.11	0.05	1.97	0.41	0.51	0.09		1.96	0.07	858	461	113.0	1.15	
34	LGA	GVL	0.50	0.50	-1.17	0.17	1.19	0.41	0.18	0.09		1.19	0.05	1511	266	98.5	1.23	
35	LGA	GVL	0.25	0.75	-0.66	0.04	1.14	0.41	0.04	0.10		1.14	0.04	1978	126	89.3	0.97	
36	LGA	THF	0.90	0.10	14.36	2.12	7.71	0.41	0.55	0.09		7.70	1.00	484	1077	190.6	1.59	30 ns Prod (15 ns/15 ns)

Figure 2: Snapshot of final data shown in `MAIN_dir/excel_spreadsheet/final_data.xlsx`

The data in `final_data.xlsx` was used to generate multilinear correlation as described in the main text. Note that the values in `final_data.xlsx` are not rescaled. Multilinear correlations can be generated in Python or MATLAB. Code for multilinear correlations are not shown here.