tsscds2018: a code for automated discovery of chemical reaction mechanisms and solving the kinetics

Aurelio Rodríguez[1], Roberto Rodríguez-Fernández[2], Saulo A. Vázquez[2], George L. Barnes,[3] James J. P. Stewart,[4] Emilio Martínez-Núñez*[2]

[1]Galicia Supercomputing Center (CESGA), Santiago de Compostela, Spain

[2]Departamento de Química Física, Facultade de Química, Campus Vida, Universidade de Santiago de Compostela, 15782, Santiago de Compostela, Spain.

[3]Department of Chemistry and Biochemistry, Siena College, 515 Loudon Road, Loudonville, NY, United States

[4]Stewart Computational Chemistry, 15210 Paddington Circle, Colorado Springs, CO 80921 USA

Corresponding Author: Emilio Martínez-Núñez (emilio.nunez@usc.es)

ABSTRACT: A new software, called tsscds2018, has been developed to discover reaction mechanisms and solve the kinetics in a fully automated fashion. The program employs algorithms based on Graph Theory to find transition state (TS) geometries from accelerated semiempirical dynamics simulations carried out with MOPAC2016. Then, the obtained TSs are connected to the corresponding minima and the reaction network is obtained. Kinetic data like populations vs time or the abundancies of each product can also be obtained with our program thanks to a Kinetic Monte Carlo routine that solves the master equation. Highly accurate ab initio potential energy diagrams and kinetics can also be obtained thanks to an interface with Gaussian09.

Keywords: Accelerated dynamics simulations, Graph Theory, reaction network, Kinetic Monte Carlo, automated method.

Program summary

Program title: tsscds2018

Licensing provisions: GNU General Public License 3 (GPL)

Programming language: Bash shell scripting, Fortran 90 and Python2.

Supplementary material:

No. of lines in distributed program, including test data, etc.: 35976

No. of bytes in distributed program, including test data, etc.: 27 Mb

Distribution format: tar.gz

Computer: All Linux based workstations

Operating system: Linux

Has the code been vectorized or parallelized?: No

Nature of problem:

Unraveling the mechanisms of chemical reactions lies at the heart of chemistry. The traditional approach to obtain reaction mechanisms from the computational side is to use chemical intuition. However, that approach can lead to an incomplete picture of the chemical problem at hand, which might be a major drawback if we fail to optimize kinetically-relevant structures.

On the other hand, computational studies of chemical reaction mechanisms often lack kinetic information, which is, many times, the only kind of results provided in the experiments. For that reason, it is crucial to develop theoretical models and methods to carry out kinetic calculations routinely.

Solution method:

Our computer program was designed to deal with the above problem without any human intervention. Starting from a given structure of our system or even from its chemical formula, tsscds2018 builds the reaction network by running accelerated direct dynamics simulations, which are analysed with tools from Graph Theory. The obtained network of minima and transition states is fed into a Kinetic Monte Carlo simulator to provide populations of all chemical species as a function of time, for the desired experimental conditions..

1. Introduction

XXX

2. The method

The method, also named tsscds, has been recently developed by one of the authors [1, 2], and it has been devised to find transition states (TSs) or, more precisely, first order saddle points in a molecular system. The basic idea behind tsscds is to run accelerated (high-temperature or high-energy) semiempirical direct dynamics simulations to break/form new bonds within the first few hundred femtoseconds. Then, an efficient post-processing algorithm identifies geometries with partly formed/broken bonds, which serve as guess structures for transition state optimizations. Once the TSs are optimized, a reaction network can be constructed by computing the intrinsic reaction coordinates (IRCs) [3], which connect TSs with minima. The method employs two levels of theory: semi-empirical and ab initio/DFT. The semi-empirical calculations are performed to run the direct dynamics and to obtain approximate TSs structures, while a higher level of theory is used to re-optimize the TSs and run IRC calculations. Two different electronic structure programs are employed: MOPAC2016 [4] and Gaussian09 [5] for the semi-empirical and ab initio/DFT calculations, respectively

Once, the fully connected stationary points are obtained, rate coefficients for each elementary step are calculated from statistical theories [1, 2], and the kinetics are solved using Kinetic Monte Carlo (KMC) [6].

3. Structure of the program

The program finds reaction pathways and solves the kinetics at two levels of theory, as mentioned above. Two scripts, **llcalcs.sh** and **hlcalcs.sh**, have been written to carry out all the low-level (ll) and high-level (hl) calculations, respectively. Each of them, in turn are made up of different modules/programs (written in Bash shell scripting or Python2) to carry out specific tasks.

Figure 1 shows a flowchart of **llcalcsh.sh**. As seen in the figure, the script has several components. It starts with a loop, shown on the left, that will be carried out for a given number of cycles or iterations (*niter*). The loop starts executing **tsscds_parallel.sh**, which submits a number of parallel and independent accelerated dynamics simulations (*ntasks*) using MOPAC2016 [4]. Figure 2 shows an example of a **tsscds_parallel.sh** job consisting of 4 parallel tasks, each of them performed by a script called **tsscds.sh**. In turn, the first step of **tsscds.sh** consists of selecting initial Cartesian coordinates **q** and momenta **p** using a microcanonical or a canonical ensemble, which is done by either **nm.exe** or **termo.exe**, respectively. Both programs are written in fortran90. The initial energy or temperature of the system is chosen automatically by the program (the reader is referred to the tutorial for details). Having **q** and **p** being chosen, **tsscds.sh** runs now a number of trajectories (*ntraj*) using a locally modified version of DRC module in MOPAC2016. Details of the modified DRC module are given in the tutorial. Once the accelerated dynamics calculations are completed, **bbfs.exe** (written in fortran90) locates guess transition states structures from the geometries along the trajectories [1]. Finally, MOPAC2016 optimizes the transition states using the standard Eigenvector Following algorithm.

After tsscds_parallel.sh has completed all (parallel) tasks, **irc.sh** screens the obtained structures to remove possible redundancies and/or saddle points associated to van der Waals intermediates, which are of little importance in a kinetic study (see Figure 1). Following completion of the screening, IRC calculations are carried out in both the forward and backward directions [3].

The last points of each IRC are the initial guesses of subsequent optimizations carried out by **min.sh**, a procedure whereby each TS can be connected with the corresponding minimum energy structures. Thus, a reaction network is built, and each structure is labelled as either intermediate, or product (containing several fragments). Additionally, groups of conformational isomers are identified, which is very useful to carry out coarse grained kinetics simulations as discussed below. The construction of the reaction network and labelling of the different structures are performed by **rxn_network.sh** script.

As seen in Figure 1, **rxn_network.sh** closes the loop, and its output is fed into **tsscds_parallel.sh**. In particular, the newly generated minima are needed by **tsscds_parallel.sh** because the ensembles of trajectories are initialized not only from the starting minimum energy structure but also from the new minima. When a maximum number of iterations is reached, the kinetics is solved using **kmc.sh** (see Figure 1), which employs Kinetic Monte Carlo [6] (KMC) simulations. This script calculates rate coefficients for every single elementary step and employs a fortran90 KMC program to solve the master equation. Finally, **final.sh** gathers all relevant mechanistic and kinetics information obtained throughout the calculations.

As mentioned above, the reaction network and kinetics results can also be obtained using an ab initio/DFT level of theory with G09. The hl tasks are performed with **hlcalcs.sh**, which is

the counterpart of **llcalcs.sh** described in the previous paragraphs. Since the TSs have been already found at low-level, the ll structures are now the initial guesses for the hl optimization. Additionally, the product fragments are now optimized to construct more accurate potential energy diagrams. Therefore, the structure of **hlcalcs.sh** is somehow different from that of **llcalcs.sh**, as seen in Figure 3. Specifically, the different tasks carried out by each component of **hlcalcs.sh** are: 1) hl optimization of the TSs obtained at the ll (**TS.sh**); 2) hl IRC calculations from the TSs optimized in the previous step (**IRC.sh**); 3) hl optimization of the corresponding intermediates (**MIN.sh**); 4) construction of the hl network (**RXN_NETWORK.sh**); 5) kinetics simulations on the hl network (**KMC.sh**); 6) hl optimization of the products (**PRODs.sh**); and 7) Gathering of the important mechanistic and kinetic results (**FINAL.sh**).

4. Discovering the reaction mechanisms of formic acid.

This section deals with the use of the program using a simple example. Specifically, we have chosen the dissociation of formic acid (FA) as a test case. The reader is referred to the tutorial that comes with this distribution for detailed instructions to install the program as well as for a thorough explanation of the program execution and input/output files.

The program is strongly dependent on a number of other tools/packages, which must be installed in the linux distribution: Environment Modules, G09, GNU Parallel, Python2 (with the Numpy and Scipy libraries), SQLite3, and Zenity3.

Environment Modules is a tool that allows the user to load the *tsscds* environment variables only when the program is employed, instead of initializing the environment when they log in.

G09 is needed to run the hl calculations. If it is not available, only the ll results can be obtained with *tsscds*; work is in progress to interface *tsscds* with other electronic structure codes.

GNU Parallel is employed to run tasks in parallel, and it is employed by several of the components of **llcalcsh.sh** and **hlcalcs.sh** that we have seen above: **tsscds_parallel.sh**, **irc.sh**, **min.sh**, **TS.sh**, **IRC.sh**, **MIN.sh** and **PRODs.sh**. This tool allows *tsscds* to run in a linux workstation using multiple processors. Additionally, the different scripts can be submitted to a Slurm job scheduling system as explained in the tutorial.

Although most of the scripts have been written in bash, a number of them are written in Python2, which must be installed alongside with its Numpy and Scipy libraries.

SQLite3 is needed because the optimized geometries, frequencies, energies, and chemical formulas of all structures (intermediates, TSs and products), as well as G09 input files are stored in SQLite tables.

Finally, Zenity3 is employed to create interactive dialog boxes to input the data, which makes the scripts more user-friendly.

Having installed the above packages and *tsscds* itself (instructions are given in the tutorial), all calculations for FA can now be carried out. Input files FA.xyz and FA.dat, which must be copied into your working directory (named FA for instance), are available in the examples folder that comes with this distribution.

4.1. Description of the input files. Only two input files (*FA.xyz* and *FA.dat*) are needed to run this example.

- File *FA.xyz*, where FA is the name of the system, contains the Cartesian coordinates of the system, usually the most stable conformer of the reactant molecule.
- File *FA.dat* contains all parameters of the calculation (see Figure 4), and it will be explained in detail in the following.

This file is split in 5 different sections. Each line, within each of the sections, starts with a (case sensitive) keyword, followed by some values or keywords.

In the General section, the user provides details of the electronic structure calculations. Several keywords are allowed in this section, and they will be explained in the following.

- *molecule* refers to the name of the system.
- *LowLevel* is any of the semiempirical methods implemented in MOPAC2016, with PM7 being the default, which makes this keyword unnecessary if PM7 is the choice.
- *Highlevel* is the level of theory employed in the high-level calculations.
- *HL_rxn_network* indicates whether the hl reaction network is calculated starting from all the obtained ll TSs (and the keyword should be followed by *complete*), or whether bimolecular reactions are removed, in which case *reduced* should be employed instead.
- *charge* is the charge of the system.
- *mult* is the multiplicity of the system.

The next section is called CDS (for Chemical Dynamics Simulations). Here the user provides details of the accelerated dynamics simulations. The keywords that can be employed in this section are explained in the following.

- *sampling* should be followed by any of the four possible keywords: *microcanonical*, *canonical*, *association* and *external***.** The options *microcanonical* and *canonical* refer to

the type of sampling employed to select the initial **q** and **p** for the accelerated semiempirical dynamics simulations. The other two options are explained in the tutorial.

- *ntraj* refers to the number of trajectories.

BBFS (Bond Breaking/Formation Search) section deals with the selection of structures from the trajectory results [1]. The user can select only those TS structures with imaginary frequencies greater than a given value (e.g., 200 cm^{-1} like in Figure 4) using keyword *freqmin*.

In the "structure screening section", *tsscds* screens the obtained TS structures. In particular, due to the parallel execution of the program, some of these structures might be redundant. Furthermore, other TSs may correspond to floppy van der Waals complexes formed upon fragmentation, and therefore they are of negligible importance in the kinetics. To avoid or minimize repeated structures and van der Waals complexes, the program includes a screening tool that employs Spectral Graph Theory to calculate the following quantities: SPRINT coordinates,[7] degrees of each vertex and eigenvalues of the Laplacian matrix.[2] Comparing these values (including the energy) for two structures, the mean absolute percentage error (MAPE) and the biggest absolute percentage error (BAPE) are obtained. The keywords *avgerr* and *bigerr* refer to the maximum values for MAPE and BAPE, respectively. If both the MAPE and BAPE values calculated for two structures are below *avgerr* and *bigerr*, respectively, the structures are regarded as equal.

The last keyword, called *thdiss*, refers to the eigenvalues of the Laplacian (EL). In Spectral Graph Theory, the number of 0 eigenvalues provides the number of connected graphs, which is translated here as the number of fragments in the molecular system. The keyword *thdiss* refers to the threshold for an EL to be considered 0. For instance, in our example, if an EL<0.1

(see Figure 4), then, this EL is set to 0. This keyword is used to identify van der Waals complexes that are formed upon unimolecular fragmentation.

In the Kinetics section, the user provides details for the kinetics calculations that simulate the experimental conditions. The accepted keywords in this section are explained in the following.

- *Rate* can be *canonical* or *microcanonical*, which means that the rate constants will be calculated according to Transition State Theory (TST) or Rice-Ramsperger-Kassel-Marcus (RRKM) theory, respectively.
- *EKMC* is the excitation energy (in kcal/mol) for the calculation of the microcanonical rates if the choice for rate was microcanonical.
- *TKMC* is the temperature (in K) for the calculation of the thermal (canonical) rates if the choice for rate was canonical. At present, temperatures below 100 K are not allowed.

4.2. Running the dynamics in a single processor.

Even though for production runs one should employ a single script (**llcalcsh.sh**) as described in section 4.5 below, for the sake of completeness, this and the next sections describe how to run each component of **llcalcs.sh** separately.

After loading tsscds module, a single-processor exploratory (accelerated) dynamics simulations (ten trajectories long) can be run using:

```
tsscds.sh FA.dat >tsscds.log &
```

The output file tsscds.log provides information about the calculations. The script **tsll_view.sh** can be used to check the transition states that have been found, which outputs something like what is shown in Figure 5 (since random number seeds are clock-dependent, in

general, outputs differ). The first column in the figure represents the order of appearance of each TS, the second is the filename of the MOPAC TS optimization (located in the newly created tsdirLL_FA directory), the third is the imaginary frequency (in cm^{-1}), the fourth one is the heat of formation in kcal/mol, and the next four numbers correspond to the four lowest vibrational frequencies (in cm^{-1}). The last two columns are the trajectory number from which the structure was selected, and the name of the folder where the accelerated dynamics was run.

Any compatible visualization program (e.g., Molden) can be employed to check each structure and/or to watch the animation of trajectories:

```
molden tsdirLL_FA/ts1_FA.out
molden coordir/FA_dyn1.xyz
```

4.3. Running the dynamics in multiple processors.

The exploratory dynamics can also be run in parallel using **tsscds_parallel.sh**. For instance, to interactively submit a total of 50 trajectories split in 5 different parallel tasks (10 trajectories each) the following should be employed:

```
tsscds_parallel.sh FA.dat 5
```

This will create temporary directories batch1, batch2, batch3, batch4 and batch5. The TSs found in each individual task will be copied in the same folder, tsdirLL_FA. The use of **tsscds_parallel.sh** script is only recommended for checking purposes since it runs interactively (with a Zenity progress bar), and particularly to carry out the screening. As mentioned above, production runs should employ **llcalcs.sh** script, which is described below.

4.4. Analyzing the dynamics results.

The **irc.sh** script, mentioned above, can be used to perform an initial screening of the TS structures before running the IRC calculations. The screening is invoked using *screening* as the sole argument:

```
irc.sh screening
```

This way, only a screening of the structures is carried out. The screening process, as mentioned above, involves the use of tools from Spectral Graph Theory and utilizes the three threshold values indicated above: *avgerr*, *bigerr* and *thdiss*. The names of the redundant and fragmented structures are printed on screen as well as in the file screening.log. The tutorial describes in detail the steps that need to be taken to analyze the results printed in screening.log.

Once the screening is satisfactory, IRC calculations, optimization of the minima, and the construction of the reaction network can be done using **irc.sh**, **min.sh** and **rxn_network.sh**, respectively.

At this point, we can grow the TS list by running more trajectories (with **tsscds_parallel.sh**) that will start from the newly generated minima as well as from the main structure, specified in *FA.xyz* file. This entails starting a second iteration of the loop shown on the left of Figure 1. After a given number of iterations is completed, and the number of TSs is converged, the kinetics is solved and the relevant mechanistic and kinetic information gathered using **kmc.sh** and **final.sh**, respectively. The latter script creates a new directory (FINAL_LL_FA) with the relevant information.

4.5. Running all low-level calculations using a single script

As mentioned above, for production runs the use of a single script is highly recommended. Once the screening is carried out, *i.e.*, the values for *avgerr*, *bigerr* and *thdiss* have been set, one should employ **llcalcs.sh** script to run all ll calculations. This can be accomplished by using the following:

```
nohup llcalcs.sh FA.dat ntasks niter runningtasks >llcalcs.log 2>&1 &
```

where *ntasks* is the number of tasks for **tsscds_parallel.sh**, *niter* is the number of iterations in the loop of Figure 1, and *runningtasks* is the number of simultaneous tasks. The script can be run without the arguments, and two Zenity3 dialog boxes will help you enter the arguments.

4.6. Running the high-level calculations

Once the low-level calculations have been completed, the user can go on by performing the high-level computations, which currently employ the G09 program. These calculations include the optimization of TSs, IRC calculations, optimization of minima and products, construction of the reaction network, calculation of rate coefficients and evaluation of the time evolution of the chemical species involved in the global reaction mechanism. All these steps can be performed using a single script **hlcalcs.sh**, employing the following sentence (for the FA example):

```
nohup hlcalcs.sh FA.dat runningtasks >hlcalcs.log 2>&1 &
```

As for the low-level calculations, the argument *runningtasks* is the maximum number of tasks that can be run simultaneously in your computer.

4.7. Directory tree structure of the working directory

Figure 6 shows the directory tree structure of the working directory. Folders batch1, batch2, and so on, include a coordir directory, which contains the individual trajectories computed in the associated task. The directories shown in blue will be preserved at the end of the calculations, while the other ones are temporary. The tsscds_parallel-logs directory contains a series of files that provide information on CPU time consumption for the different calculation steps.

4.8. Relevant information

As mentioned above, **final.sh** (or its high-level counterpart **FINAL.sh**) gathers all relevant information in a folder named FINAL_XL_FA (where XL=HL,LL for high-level and low-level, respectively). These folders contain the following files/folders:

Energy_profile.gnu is a file in gnuplot format with information on the relevant paths at the simulated conditions (see Figure 7 for the plot obtained for FA at the low-level). Here relevant refers to those paths that contribute to at least 0.1% of the total number of KMC simulation steps. The reader is referred to the tutorial if the minimum percentage to define a relevant path needs to be changed.

MINinfo contains information of the minima. Figure 8 shows an example of the *MINinfo* file obtained at ll for the FA test case. DE in the figure refers to the energy relative to that of the main structure specified in the *FA.dat* file (optimized with the semiempirical Hamiltonian). The integers are used to identify, independently, minima and transition states. Notice that, in this example, MIN 2 corresponds to the structure specified in *FA.xyz*.

TSinfo is similar to *MINinfo*, but this one contains the corresponding information of the TSs.

table.db is a SQLite3 table containing the geometries, energies and frequencies of minima, products and TSs, respectively; *table* is a tag that can be either *min*, *prod* or *ts*. The different properties can be obtained using **select.sh** script, which should be run in the FINAL_XL_FA folder:

```
select.sh property table label
```

where *property* can be: *natom*, *name*, *energy*, *zpe*, *g*, *geom*, *freq*, *formula* (only for prod) or *all*, and label is one of the numbers shown in RXNet (see below), which are employed to label each structure. At the semiempirical level, the energy values correspond to heats of formation. For high-level calculations, the tables collect the electronic energies.

RXNet contains information of the reaction network. Figure 9 shows an example of the *RXNet* file obtained at ll for the FA test case. As can be seen, for each transition state, this file specifies the associated minima and/or products and their corresponding identification numbers. Notice that TSs, minima (MIN) and products (PROD) have independent identification numbers. The chemical formulas of the products fragments are listed at the end of the file.

RXNet.cg is similar to *RXNet* with the extension cg standing for “coarse-grained”. By default the KMC calculations are “coarse-grained”, that is, conformational isomers form a single state, which is taken as the lowest energy isomer [2]. Such reaction network, which also removes bimolecular channels, is shown in Figure 10 for the FA test case, calculated at the ll.

The last column of Figure 10 with the flag “CONN” or “DISCONN” indicates whether the given process is connected with the others (CONN) or whether it is isolated (DISCONN). This flag is useful when the chosen starting intermediate for the KMC simulations is other than the starting structure, because the selected intermediate has to be connected with the others.

kineticsFvalue contains the kinetics results, namely, the final branching ratios and the population of every species as a function of time. In the name of the file, F is either "T" or "E" for temperature or energy, respectively, and "value" is the corresponding (temperature or energy) value. For instance, the kinetics results for a canonical calculation at 298 K would be printed in a file called kineticsT298.

populationFvalue.gnu is a file in gnuplot format containing the population of each species as a function of time. Figure 11 shows the plot obtained with this file for the decomposition of FA using the PM7 stationary points.

normal modes is a folder that contains the normal mode eigenvectors and eigenvalues of TSs and minima. They are specified in Molden format, for visualization with this graphic software.

4.9. Details of the kinetics simulations

As indicated above, by default, the KMC simulations regard conformational isomers as a single state, which speeds up the calculations [2]. However, each conformational isomer could be treated as a single state in the KMC calculations. If that is the case, the reaction network needs to be reconstructed and the kinetics simulations carried out on the extended network. That entails running again **rxn_network.sh, kmc.sh** and **final.sh**, using *allstates* as the argument for the first one:

```
rxn_network.sh allstates
kmc.sh
final.sh
```

Additionally, when the calculations seek to simulate a thermal experiment (i.e, when *rate canonical* is employed in the input file), the kinetics results can be re-run for a different

temperature from that specified in the input file through the keyword *TKMC* (see above). This can be easily done using **kinetics.sh** with the following arguments:

```
kinetics.sh temp calc (allstates)
```

where *temp* is the new temperature of the system (in K), and *calc* is either *ll* (low-level) or *hl* (high-level). As mentioned above, the argument *allstates* indicates that conformational isomers form a single state.

5. Other capabilities of the code

Besides the above basic tools, a number of other features are implemented in the program. The reader is referred to the tutorial for a thorough explanation of those other capabilities, as here only a very brief summary is provided. The other possibilities of the program refer to other sampling options (*intermolecular* and *external*) and to further accelerated dynamics choices.

The options *intermolecular* and *external* are employed to optimize intermolecular complexes and to the use of external programs to carry out the dynamics simulations, respectively. The latter option has been added to interface the chemical dynamics simulation code VENUS [8] with tsscds. This feature could be of great interest to simulate mass spectrometry experiments, where collisions with projectiles are employed to dissociate the molecule [9] .

Besides the standard values/parameters explained above, the tutorial offers a number of other choices for advanced users. Those extended options are explained in detail in the tutorial.

Finally, other accelerated dynamics techniques have been included in the program, like the use of phase space constraints [10] or bias potentials. Examples of how to use those feature with some simple examples are given in the tutorial.

6. Applications

In the previous section, and for testing purposes, the FA example was presented. However, the program has been employed to elucidate the reaction mechanisms of a number of different systems. The most relevant results obtained for these systems are summarized in the next paragraphs.

The smallest systems studied with our procedure are: formaldehyde, formic acid (FA), and vinyl cyanide (VC) [1], for which a total of 7, 12, and 83 TS structures have been located, respectively. Of significance, a new TS for the water-gas shift reaction (WGSR: $CO + H_2O \rightarrow CO_2 + H_2$) was found for FA [1]. This is an interesting result since the WGSR is bimolecular, whereas the accelerated dynamics is unimolecular, which exemplifies the highly non-statistical nature of our simulations and the wealth of information (structures) that can be drawn using our methodology. Also, the theoretical VC decomposition kinetics, studied separately [11], leads to nearly perfect agreement with the experimental HCN/HNC branching ratio.

The fragmentation kinetics of propenal is very complex with many different fragmentation channels that involve well over 250 transition states [2]. The computed branching ratios for the different dissociation channels of the molecule agree very well with the available experimental data.

Three novel HCl dissociation channels of acryloyl chloride (AC) were discovered with our procedure [12]; those channels had gone unnoticed in previous theoretical work. Furthermore, quasi-classical trajectory simulations starting from the obtained HCl dissociation TSs lead to bimodal rotational distributions of HCl, which agree very well with experiment.

The fragmentation dynamics of protonated uracil is very rich, and more than one thousand stationary points and 751 reactive channels were discovered using tsscds [9]. The predominant dissociation channels of the cation are in very good agreement with the results of mass spectrometry experiments.

Very recently, *tsscds* has been also adapted to study organometallic catalysis. In particular, the cobalt-catalyzed hydroformylation of ethylene was the chosen test case [13]. The study entailed running *tsscds* in eight different systems, which involved combinations of the starting materials (CO, H_2 and ethylene) with the catalyst $Co(CO)_3$. After merging all results, the kinetics simulations give rise to a theoretical rate law for hydroformylation that agrees rather well with the experimental one. Additionally, our method predicts that hydrogenation of ethylene is a side reaction that can be predominant under certain experimental conditions.

Our methodology can also be employed to understand the possible sources of HCN/HNC formation in astrophysical environments [14]. In particular, time-resolved infrared spectroscopy experiments detected formation of both HCN and HNC after 193-nm photolysis of methyl cyanoformate. Our automated protocol was able to locate several cyclic transition states leading to HNC and HCN, and the resulting HCN/HNC branching ratios found in our simulations are not far from those obtained in the experiments. Furthermore, quasi-classical trajectory deduced internal energy distributions of HCN and HNC are in very good agreement with the experimental

ones. The work may help explain the observed overabundance of HCN in astrophysical environments.

7. Summary and future work

A user-friendly program for the discovery of reaction mechanisms and for efficiently solving the kinetics is presented in this manuscript. The code relies on exploratory semiempirical accelerated dynamics simulations carried out by a modified version of MOPAC2016 and on an efficient geometry-based algorithm recently developed by one of the authors. The resulting potential energy diagrams and kinetics can be obtained not only at the semiempirical level, but also employing higher level (ab initio/DFT) electronic structure calculations using G09.

The only input needed from the user is a file containing some details of the calculations, and an initial input structure, that can be taken from experiments, previous computations, or even constructed with any visualization graphics software. The procedure is therefore fully automated, except for the selection of three screening parameters that serve to avoid redundant structures and transition states connecting van der Waals minima. In that case, the user is advised to check a few structures with the naked eye to judge the validity of the input screening parameters. That is the only human intervention in the process. After that, several components of the program undertake the different tasks needed to generate potential energy diagrams and plots of populations vs time.

Several new tools will be incorporated in a future version, and our team has already started working on the following features:

1) Treatment of barrierless reactions. BBFS algorithm is responsible for the identification of guess transition states that will be then subjected to the standard EF algorithm to optimize saddle points. Therefore, processes that occur without any barrier are elusive, and many times those barrierless reactions are predominant. Identification of those processes in not a major issue, but an accurate automated evaluation of the involved rate coefficients is not a trivial task.

2) A second tool that will be incorporated in future versions is the analysis of secondary fragmentations. That entails running the accelerated dynamics not only for the input molecule but also for the fragments that result upon dissociation. This feature is very important, for instance, in the theoretical analysis of mass spectrometry experiments.

3) The third feature that will be present in a future release is the analysis of bimolecular reactions. Even though one can already start from any given structure of the system, including shallow van der Waals minima, it would be desirable to start the dynamics from the separated chemical species and run the bimolecular dynamics. That entails the implementation of appropriate samplings of initial conditions and an ample selection of the initial relative velocities and energies of each fragment.

Acknowledgements

One of us (JJPS) thanks the National Institute of General Medical Sciences of the National Institutes of Health (Award Number R44GM108085) for support in this work. SAV and EMN thank Xunta de Galicia and Ministerio de Economía y Competitividad of Spain (Research Grants No ED431C 2017/17 and CTQ2014-58617-R, respectively). Authors gratefully thank

CESGA (Galicia Supercomputing Center, Santiago de Compostela, Spain) for providing access to computing facilities

Figure captions

Figure 1. Flowchart of **llcalcs.sh** script.

Figure 2. Different tasks carried out by **tsscds_parallel.sh**. In this example of total of 4 independent **tsscds.sh** jobs are carried out in parallel.

Figure 3. Flowchart of **hlcalcs.sh** script.

Figure 4. Input file *FA.dat* employed to study the decomposition of formic acid.

Figure 5. Output of **tssll_view.sh** script for the FA test case.

Figure 6. Directory tree structure of the working directory.

Figure 7. Plot corresponding to the relevant paths obtained at low-level for FA. The labelling and/or number of structures depends on the number of iterations, trajectories, and might differ in general from those obtained in separate runs because of the different random number seeds employed in the dynamics.

Figure 8. File *MINinfo* corresponding to the FA test case. The ll results are shown here.

Figure 9. File *RXNet* corresponding to the FA test case. The ll results are shown here.

Figure 10. File *RXNet.cg* corresponding to the FA test case. The ll results are shown here.

Figure 11. Population of every chemical species (with non-zero population) vs time for the decomposition kinetics of FA evaluated using the PM7 stationary points.

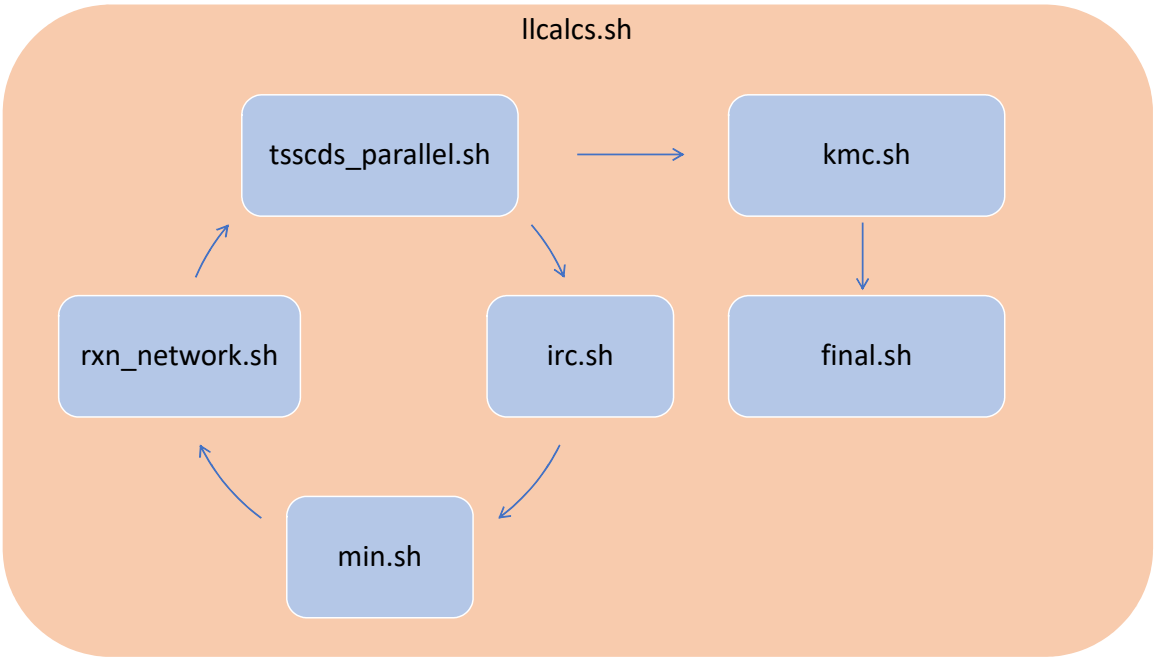

llcalcs.sh
tsscds_parallel.sh
kmc.sh
rxn_network.sh
irc.sh
final.sh
min.sh

Figure 1

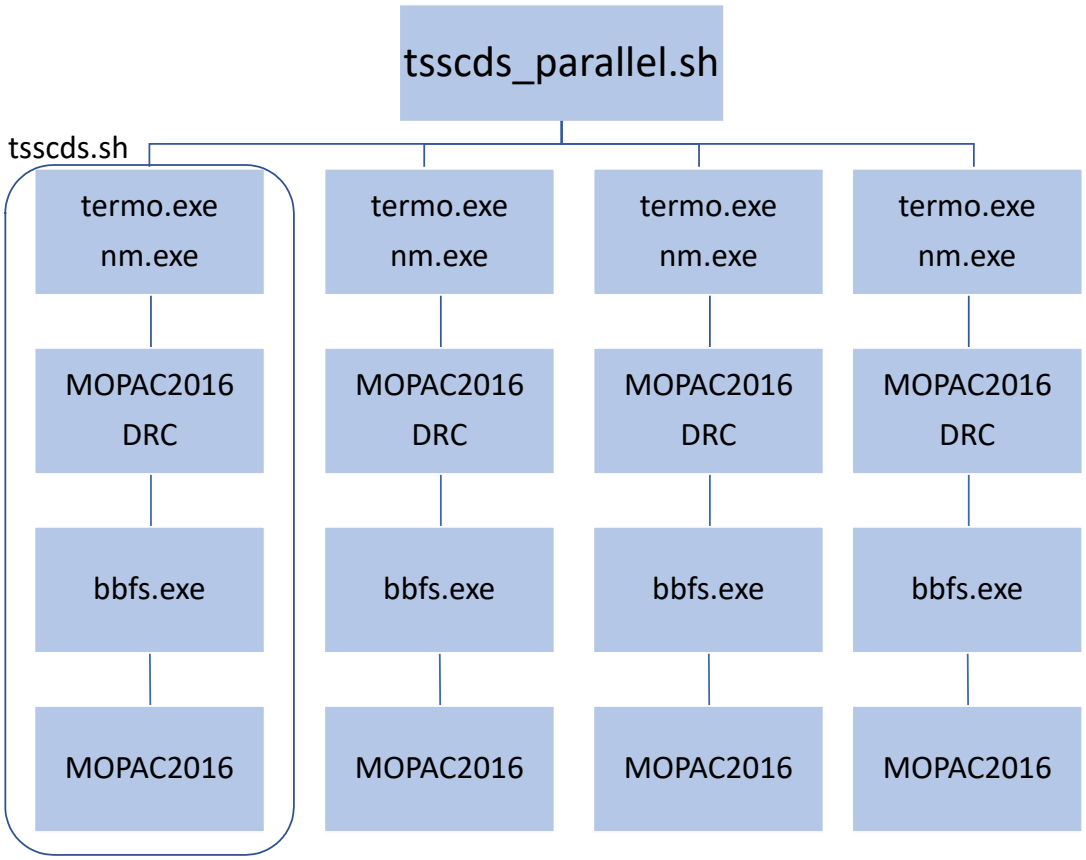

tsscds_parallel.sh
tsscds.sh
termo.exe
nm.exe
MOPAC2016
DRC
bbfs.exe
MOPAC2016
termo.exe
nm.exe
MOPAC2016
DRC
bbfs.exe
MOPAC2016
termo.exe
nm.exe
MOPAC2016
DRC
bbfs.exe
MOPAC2016
termo.exe
nm.exe
MOPAC2016
DRC
bbfs.exe
MOPAC2016

Figure 2

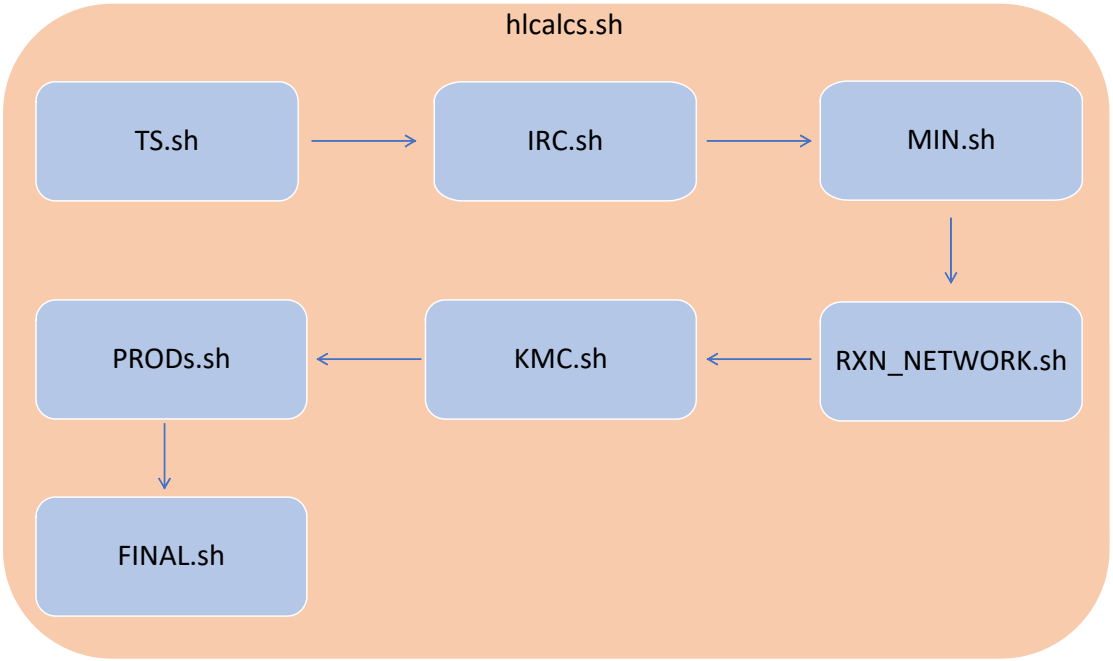

hlcalcs.sh
TS.sh
IRC.sh
MIN.sh
RXN_NETWORK.sh
KMC.sh
PRODs.sh
FINAL.sh

Figure 3

```
--General section--
molecule FA
HighLevel b3lyp/6-31G(d,p)
HL_rxn_network complete
charge 0
mult 1

--CDS section--
sampling microcanonical
ntraj 10

--BBFS section--
freqmin 200

--Screening of the structures section--
avgerr 0.008
bigerr 2.5
thdiss 0.1

--Kinetics section--
Rate  microcanonical
EKMC  150
```

Figure 4

ts #	MOPAC file name	w_imag	Energy	w1	w2	w3	w4	traj #	Folder
1	ts1_FA	1587.3i	-35.71	204.3	438.3	461.3	726.8	1	FA
2	ts2_FA	2009.6i	-17.61	327.2	472.7	522.7	1078.6	2	FA
3	ts3_FA	2930.8i	-20.17	450.6	586.9	908.6	997.2	7	FA

Figure 5

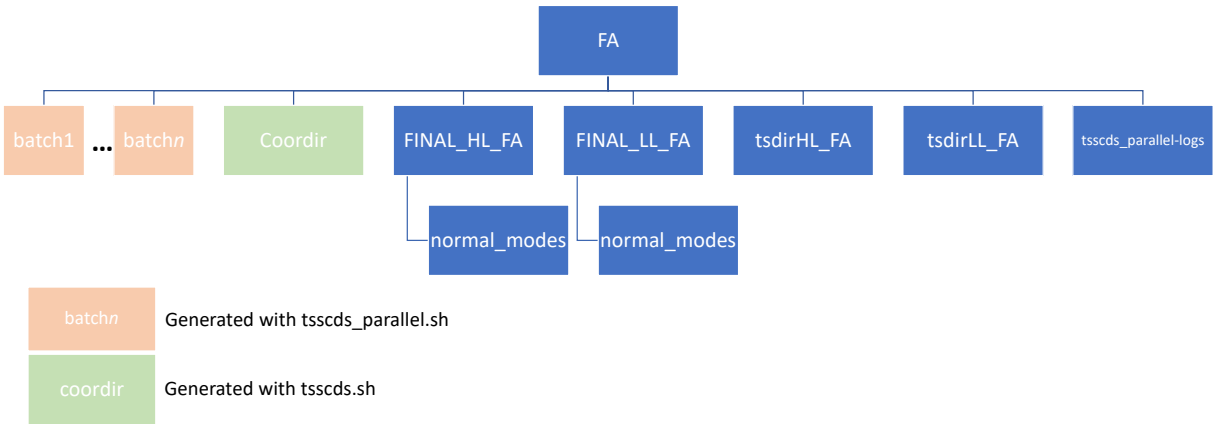

FA
batch1
...
batchn
Coordir
FINAL_HL_FA
FINAL_LL_FA
tsdirHL_FA
tsdirLL_FA
tsscds_parallel-logs
normal_modes
normal_modes
batchn
Generated with tsscds_parallel.sh
coordir
Generated with tsscds.sh

Figure 6

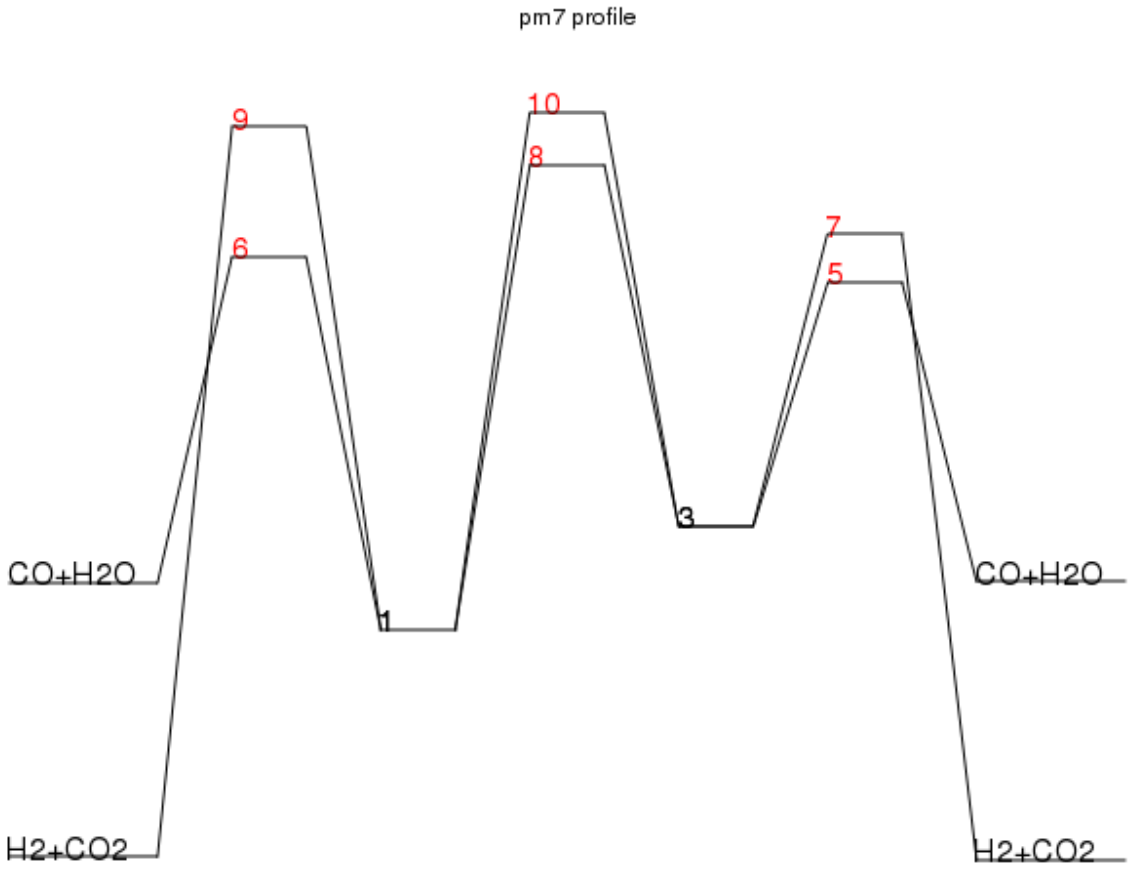
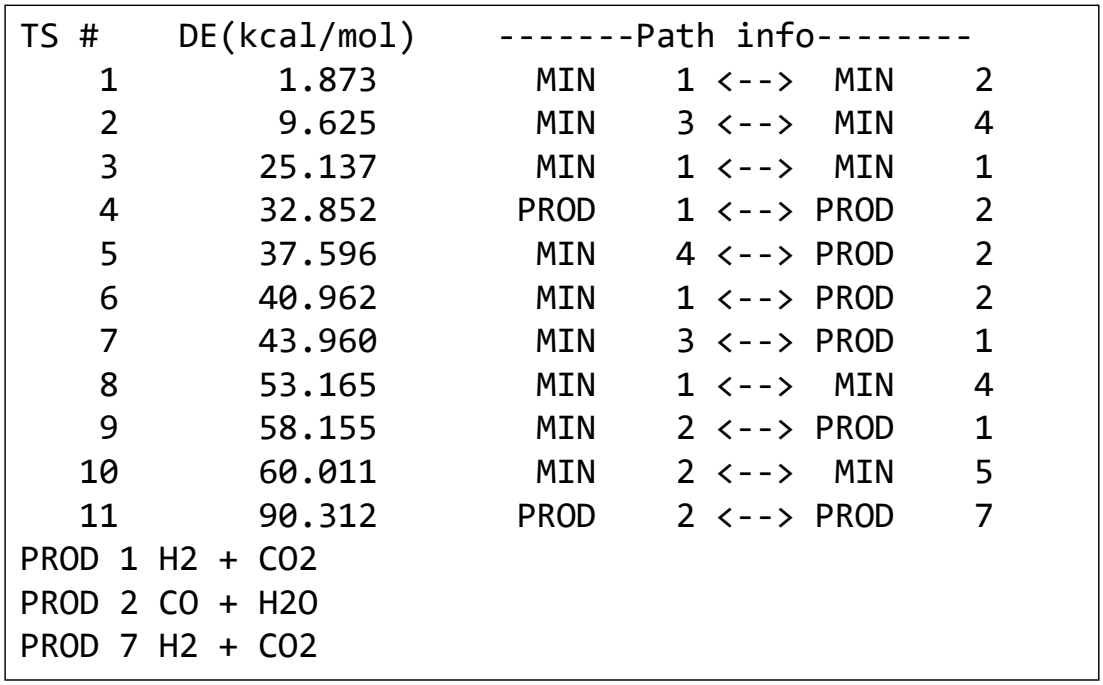


Figure 7

```
MIN #    DE(kcal/mol)
    1       -8.340
    2        0.000
    3        5.283
    4        6.710
    5       15.338

Conformational isomers are listed in the same line:
1 2
3 4 5
```

Figure 8



```
TS #    DE(kcal/mol)    -------Path info--------
    1         1.873        MIN    1 <-->  MIN    2
    2         9.625        MIN    3 <-->  MIN    4
    3        25.137        MIN    1 <-->  MIN    1
    4        32.852       PROD    1 <--> PROD    2
    5        37.596        MIN    4 <--> PROD    2
    6        40.962        MIN    1 <--> PROD    2
    7        43.960        MIN    3 <--> PROD    1
    8        53.165        MIN    1 <-->  MIN    4
    9        58.155        MIN    2 <--> PROD    1
   10        60.011        MIN    2 <-->  MIN    5
   11        90.312       PROD    2 <--> PROD    7
PROD 1 H2 + CO2
PROD 2 CO + H2O
PROD 7 H2 + CO2
```

Figure 9

```
TS #    DE(kcal/mol)    -------Path info--------
    5        37.596        MIN    3 <--> PROD    2    CONN
    6        40.962        MIN    1 <--> PROD    2    CONN
    7        43.960        MIN    3 <--> PROD    1    CONN
    8        53.165        MIN    1 <-->  MIN    3    CONN
    9        58.155        MIN    1 <--> PROD    1    CONN
   10        60.011        MIN    1 <-->  MIN    3    CONN
PROD 1 H2 + CO2
PROD 2 CO + H2O
PROD 7 H2 + CO2
```

Figure 10

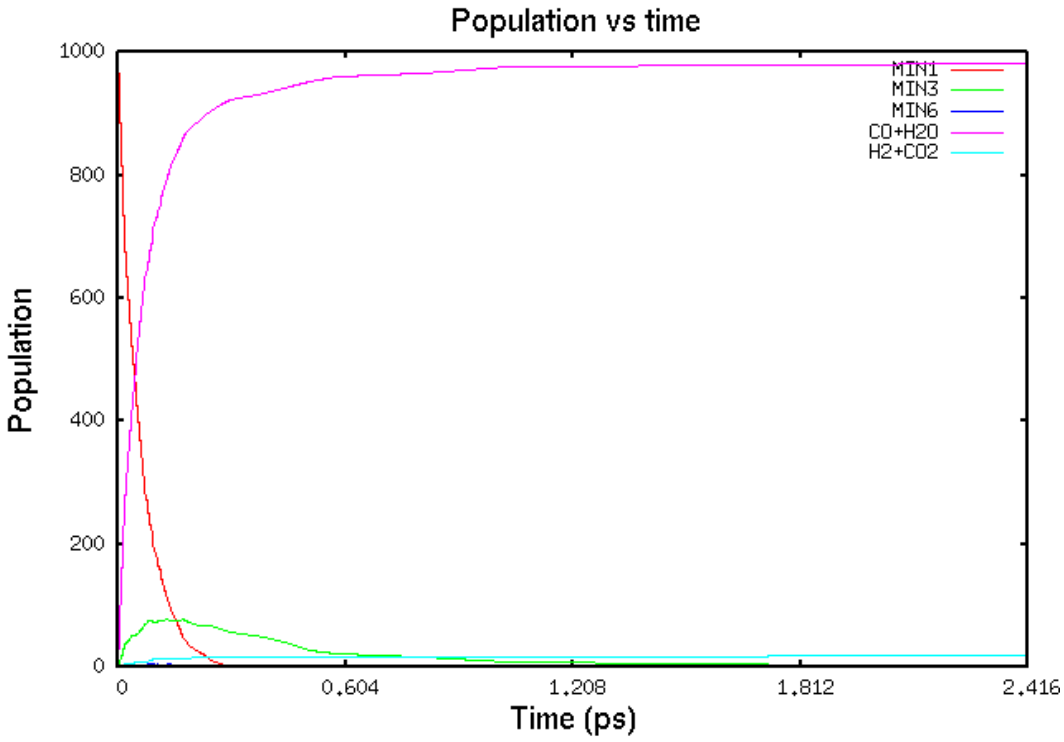

Population vs time
Population
Time (ps)
MIN1
MIN3
MIN6
CO+H2O
H2+CO2
0
200
400
600
800
1000
0
0.604
1.208
1.812
2.416

Figure 11

REFERENCES

[1] E. Martínez-Núñez, An automated method to find transition states using chemical dynamics simulations, J. Comput. Chem., 36 (2015) 222-234.
[2] E. Martínez-Núñez, An automated transition state search using classical trajectories initialized at multiple minima, Phys. Chem. Chem. Phys., 17 (2015) 14912-14921.
[3] K. Fukui, The Path of Chemical Reactions-The IRC Approach, Acc. Chem. Res., 14 (1981) 363.
[4] J.J.P. Stewart, MOPAC2016, Stewart Computational Chemistry, Colorado Springs, CO, USA, HTTP://OpenMOPAC.net, 2016.
[5] M.J. Frisch, G.W. Trucks, H.B. Schlegel, G.E. Scuseria, M.A. Robb, J.R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G.A. Petersson, *et. al.* Gaussian 09 Gaussian Inc., Wallingford CT, 2009.
[6] D.T. Gillespie, A general method for numerically simulating the stochastic time evolution of coupled chemical reactions, J. Comput. Phys., 22 (1976) 403-434.
[7] F. Pietrucci, W. Andreoni, Phys. Rev. Lett., 107 (2011) 085504.
[8] W.L. Hase, K. Bolton, P.d. Sainte Claire, R.J. Duchovic, X. Hu, A. Komornicki, G. Li, K.F. Lim, D.-H. Lu, G.H. Peslherbe, et. al. Venus05, a general chemical dynamics computer program, 2004.
[9] E. Rossich Molina, J.-Y. Salpin, R. Spezia, E. Martinez-Nunez, On the gas phase fragmentation of protonated uracil: a statistical perspective, Phys. Chem. Chem. Phys., 18 (2016) 14980-14990.
[10] E. Martinez-Nunez, D.V. Shalashilin, Acceleration of classical mechanics by phase space constraints, J. Chem. Theor. Comput., 2 (2006) 912-919.
[11] S.A. Vazquez, E. Martinez-Nunez, HCN elimination from vinyl cyanide: product energy partitioning, the role of hydrogen-deuterium exchange reactions and a new pathway, Phys. Chem. Chem. Phys., 17 (2015) 6948-6955.
[12] R. Perez-Soto, S.A. Vazquez, E. Martinez-Nunez, Photodissociation of acryloyl chloride at 193 nm: interpretation of the product energy distributions, and new elimination pathways, Phys. Chem. Chem. Phys., 18 (2016) 5019-5026.
[13] J.A. Varela, S.A. Vazquez, E. Martinez-Nunez, An automated method to find reaction mechanisms and solve the kinetics in organometallic catalysis, Chem. Sci., 8 (2017) 3843-3851.
[14] M.J. Wilhelm, E. Martínez-Núñez, J. González-Vázquez, S.A. Vázquez, J.M. Smith, H.-L. Dai, Is Photolytic Production a Viable Source of HCN and HNC in Astrophysical Environments? A Laboratory-based Feasibility Study of Methyl Cyanoformate, The Astrophysical Journal, 849 (2017) 15.