

Supplementary Information:

Chatbot-Assisted Quantum Chemistry for Explicitly Solvated Molecules

Rohit S. K. Gadde,¹ Sreelaya Devaguptam,¹ Fangning Ren,¹ Rajat Mittal,²

Lechen Dong,¹ Yao Wang,¹ Fang Liu^{1*}

¹Department of Chemistry, Emory University, Atlanta, GA 30322, United States

²Department of Physics and Astronomy, Clemson University, Clemson, SC 29631, United States

Table of Contents

Text S1. The mechanism for the intent-based conversations in AutoSolvateWeb	S2
Text S2. Scaling analysis	S4
Text S3. Implicit solvent model results for the solvatochromism lab.	S4
Text S4. Example conversation for the solvatochromism lab.	S5
Text S5. Job queue	S9
Text S6. Installation guide	S10
Text S7. Example conversation between the chatbot and the user	S11
Figure S1. Conversation flow of AutoSolvateWeb chatbot	S16
Figure S2. Installation guide figure	S17
Figure S3. Job queue figures	S18
Figure S4. An example conversation with an intent match for step3.start	S20
Figure S5. An example page setting with allowed intent routes	S21
Figure S6. Two example conversations showing intent match only happen in scope	S22
Table S1. All intents and their training phrases	S23
Table S2. List of all solvents and their force fields in AutoSolvateWeb	S25
Table S3. Description of all files in the AutoSolvateWeb output ZIP	S25
Table S4. Step-1 runtime for selected solutes and solvents	S32
Table S5. List of labels measured as part of load testing	S35
Table S6. Performance results of each load testing	S35
Table S7. Load testing: the max number of users for the label ‘confirmParameters’	S36
Table S8. Load testing: the max number of users that can load the web interface.	S36
Table S9. List of all pages used in the conversation of Autosolvate chatbot	S37

* Electronic mail: fang.liu@emory.edu

Text S1. The mechanism for the intent-based conversations in AutoSolvateWeb

The **Method** section succinctly describes the conversation in DialogFlow CX as a collection of flows embedded in a finite state machine (**Figure S1**). Here, we give a more detailed explanation with some examples.

Overview of the conversation flow. As mentioned in the **Result** section, the AutoSolvate command-line-based backend automates the explicit-solvent simulations in three steps (Step-1, Step-2, Step-3). For each step, the command-line interface requires the user to: (1) specify all keywords, (2) assemble the keywords into a Linux command, (3) execute that command, (4) download output. Hence, the conversation flow focuses on using the chatbot to finish these major tasks (and some associated minor tasks) one by one.

Figure S1 shows the conversation flow of the AutoSolvateWeb chatbot as a state flow diagram, where each state is called a “page”, focusing on finishing a different task. Some pages are configured to collect information from the end-user, whereas others just print messages to the user. The arrows in the diagram indicate the direction of the conversation flow. For example, the top of **Figure S1** is the Start Page, which begins the conversation. Once the user opens the chatbot, the Start Page is launched, and the chatbot greets the user with messages explaining the function of AutoSolvateWeb. Then, since the solute information is the foundation for all simulation steps, the conversation directly follows the arrow to the Solute Step page to specify the solute by specificizing the IUPAC name or by uploading a .xyz file. Hence, the flow bifurcates to the *GetSoluteName* page or the *UploadFile* page. If the user chooses to update the .xyz file (via the page *UploadFile*) and does not provide the chemical name, they will be led to the *GetSoluteCharge* page to specify the charge because the .xyz file only contains coordinates and does not imply the molecule’s net charge. Then, the flow continues to the *Use Suggested Params* page, where parameters (input keywords) are automatically populated based on the solute structure, and the chatbot asks the user to check whether they would like to use these default parameters or change them. Once all these solute-related tasks are done, the flow sequentially finishes the tasks related to Step-1 (via three pages: *Step-1*, *Step1 End*, *Proceed to Step 2*), Step-2 (via six pages: *Step-2*, *Step-2 Dry Run*, *Step-2 No Dry Run*, *Step-2 QM/MM*, *Step-2 End*, *Proceed to Step 3*), and Step-3 (via two pages: *Step-3*, *Step-3 End*). Here, the *Step-X* page (X=1, 2, or 3) collects the input parameters for that step by asking the user a sequence of questions. The *Step-X End* page asks the users to confirm the input parameters and post webhooks to the Autosolvate web server, which assembles the corresponding Linux command and executes the AutoSolvate backend to perform the simulation. Finally, the conversation flow will reach one of the ending pages: *NormalEnd*, *ErrorEnd*, and *End*. A detailed description of each page is given in **Table S9**.

Throughout the conversation flow, the chatbot always proactively asks the user questions to ensure the direction of the conversation follows the diagram (**Figure S1**). Hence, the user does not need to worry about conceptualizing the conversation to make the chatbot do the simulation. However, this does not mean that the users cannot answer questions. Users can issue instructions to the chatbot to jump to different pages in the conversation flow by triggering intents.

Intents. An *intent* categorizes an end-user's intention for one conversation turn.¹ For our AutoSolvateWeb chatbot, when an end-user enters a message, the Dialogflow CX compares the input to intent training phrases to find the best match. This process is called intent matching.¹ As mentioned in the **Method** section, we have three intents corresponding to restarting each simulation step: “step1.start” “step2.start”, and “step3.start”. At any conversation stage, if the user enters a message that fuzzy matches the training phrases of these intents, the corresponding intent will be triggered, and the dialog will be switched to the corresponding page. A complete list of the intents and training phrases defined for our chatbot is listed in **Table S1**.

For example, the intent “step3.start” is trained based on the following phrases: “run step-3”, “Run Step-3 with following inputs: id first frame:0 interval:1 step size: 10”, “Do Microsolvated cluster extraction”, “Microsolvated cluster extraction”. At any stage, if the user sends a message vaguely matching the meaning of these training phrases, the conversation will be diverted to the *Step-3* page, and the chatbot will start to ask the user questions related to the parameters of Step-3. **Figure S4** demonstrates an example conversation containing a phrase matching this intent. While the user is still at the beginning of the conversation about solutes, if they say something matching the phrase of the intent “step3.start”, they immediately turn to Step-3.

However, it is worth noting that an intent match does not happen whenever the user inputs a message. Intent matching can only occur for intents associated with an *intent route*² in *scope*². In plain language, to make the intent match happen, you must have defined some action to take associated with an intent, which is called an *intent route*. Here, the action can be creating a response (sending back a message to the user) or transitioning the current page to another page (like directly going to Step-3). In addition, intent match requires the intent route to be allowed in the current conversation context, which is controlled in the setting of each page. For example, **Figure S5** shows the settings for the *SoluteStep* page in our flow. In this example, the allowed intent routes are “DownloadRequiredFiles” and “UploadRequiredFiles”, as listed below “Routes”. Additionally, we made the three intents, “step1.start”, “step2.start”, and “step3.start”, in scope globally in our flow (**Figure S1**), so the intent matching can always happen for them on any page even if they are not explicitly listed under “Routes”.

In contrast, other intents defined in our chatbot do not have global scope and cannot always be triggered when the user inputs something matching the phrase. For example, the intent “DownloadRequiredFiles” (to choose to download the solute structure file from the PubChem database by name) is only in scope on the *SoluteStep* page. If you are currently on a different page of the conversation (e.g., setting the solute charge on the *GetSoluteName* page), then intent matching cannot happen even if you send the same message “Download the solute file from Pubchem API” (**Figure S6**).

Finally, intents allow annotations for the training phrases, which control how data is extracted from the end-user expression to populate DialogFlow parameters. As shown in **Table S1**, the “step2.start” intent has some annotated training phrases like “Do MD automation in dry-run mode and run QM/MM step”. Here, the highlighted phrases “in dry-run mode” and “run QM/MM step” are annotations. When the user inputs a message this way, it not only triggers the

intent route to restart Step-2, but also directly sets the two parameters, “dry-run mode” and “run QM/MM”, to True. In this way, the chatbot will not need to ask further questions about the two parameters, and the experienced user will save time interacting with the chatbot.

Text S2. Scaling Analysis

The runtime for Step-1 was accessed on a test set of 10 solute molecules with different input parameters (**Table S4**). The number of heavy atoms in the solute ranged from 1 to 33 (**Figure S3**). 110 test runs were timed sequentially using the job queue on the AutoSolvate server (4-core Virtual Cloud instance with AMD EPYC-Milan Processor and 15GB RAM; Nvidia A100X-8C virtual GPU, 8 GB DRAM). No other jobs were allowed while the test runs were executed. Further, only 1 test run was allowed at a time. Each solute geometry was specified as a solute geometry file (‘.xyz’ format) included in **Supplementary Data 2**. For closed-shell solutes (with spin multiplicity 1), AutoSolvate drives AmberTools as the backend to calculate the AM1-BCC atomic charges. For open-shell solutes, AutoSolvate drives the GAMESS quantum chemistry package as the backend to calculate the electrostatic potential needed for fitting the restrained electrostatic potential (RESP)³. Lastly, the GAMESS⁴ executable was configured to use 1 process on 1 node with a word limit of 128MW (with WORD size 8 Bytes).

Text S3. Implicit solvent model results for the solvatochromism lab.

The Polarizable Continuum Model (PCM) implemented in TeraChem was used to simulate the solvent effect. All parameters except the total dielectric constant (epsilon) are set as the default one in TeraChem. The structure of Reichardt's dye was optimized at PBE0/6-31G* level of theory with the dielectric constant $\epsilon=36.65$ (acetonitrile) and 33.00 (methanol), respectively. Then, the HOMO-LUMO gap was computed in different solvents with the corresponding minimized geometry, resulting in 2.420 eV in acetonitrile and 2.415 eV in methanol. The difference between these two gaps is no greater than 0.01 eV, far from reproducing the 0.40 eV change in the experimental absorption spectrum.

To comprehensively consider the response of the solvent during the excitation of the solute, we also utilized the non-equilibrium linear response PCM (neq-LR-PCM)⁵ and non-equilibrium state-specific PCM (neq-SS-PCM)⁶ to evaluate the vertical excitation energy of the first singlet excited state. The time-dependent density functional theory (TDDFT) with the Tamm-Dancoff Approximation was used to evaluate the excited state energy.⁷ Result shows that no matter which formula of PCM is used, the difference between the excitation energy in different solvents is no greater than 0.01 eV, still very far from reproducing the experimental 0.40 eV difference. The corresponding results are summarized in the “result.txt” in the “PCM” folder of **Supplementary Data 3**.

Text S4. Example conversation between the chatbot and the user for the solvatochromism lab.

Step - 1

User Open the chat box of AutoSolvateWeb

Chatbot I can automate initial structure generation for explicitly solvated systems. This includes input file preparation. Additionally automated QM/MM trajectory generation and microsolvated cluster extraction is supported for the explicitly solvated systems.

 Alright, Let's start by specifying the solute file. You can either:

 ‘I will upload a .xyz file’

 ‘Download from PubChem’

User I will Download from PubChem (user choose download option)

ChatBot Please enter the IUPAC name of the solute

User Reichardt's dye

ChatBot The solute entered is Reichardt's dye. Please confirm if the solute is correct?

User Yes, it' correct (user confirmed solute name is correct)

ChatBot You have selected Reichardt's dye as the solute. The following are the recommended solute parameters for this solute:

 Solute Charge: 0

 Solute Spin Multiplicity: 1

 Solvent Cube Size (Angstrom): 33.05

 Do you want to proceed with the above solute parameters?

User No

Chatbot Alright. You will be prompted for each parameter required for Step-1.

 What is the charge of the solute?

User 0

Chatbot A sample solute spin multiplicity is 1. What do you want to use here?

User 1

ChatBot What is the solvent? Currently, I support the following.

Water

Methanol

Chloroform

NMA

Acetonitrile

User Acetonitrile (Input “Methanol” to generate the methanol solvated clusters)

ChatBot What charge method do you want to use for force field fitting?

bcc

resp

User bcc

ChatBot Please provide the solvent box size in Angstrom.

User 45.0

ChatBot All input parameters for Step-1 are validated.

The following are the inputs given for Step -1:

Solvent: methanol

Charge Method for Force Field Fitting: bcc

Solute Charge is 0

Solute Spin Multiplicity is 1

Solvent Cube Size is 45.0 Angstrom.

Do you want to proceed with the above inputs for Step -1?

User Yes

ChatBot Job Launched. Job ID: <job-id>

{Button to show output of Step-1}

{Button to download output of Step-1}

Do you want to proceed to Step-2?

Please launch a job for Step-2 only after Step-1 Job completes!

(The user needs to wait for the job to be completed.)

User Yes

Step - 2

ChatBot Let's start with step -2: MD Automation

Would you like to run the run simulation in the dry run mode?

User

Yes

ChatBot Please provide the temperature in Kelvin.

User 300

Chatbot Please provide the temperature in bar, to equilibrate during the MM NPT step.

User 1.0

ChatBot Welcome to MD Automation (Dry Run Mode). This will only generate the commands to run MD programs without executing.

Now for the Classical MD control options. What do you want the MM minimization steps to be? The default value is 100.

User 1000

Chatbot What do you want the MM heat up steps to be?

User 1000

Chatbot What do you want the MM NVE steps to be?

User 0

Chatbot What should the MM NPT pressure equilibration steps be? By default is 100.

User 500000

Chatbot Do QM/MM?

User No

Chatbot Let me generate MD input files and job scripts.

You have entered the following inputs as parameters to run MD-Automation. Please confirm if they are right

Solute Charge: 0 Solute Spin Multiplicity: 1

Temperature: 300 K Pressure: 1.0 Bar

Classical MD control options - 1. MM minimization steps: 1000 2. MM heat up steps: 100 3. MM NPT pressure equilibration steps: 500000 4. MM NVE production run steps: 0

QM/MM control options - 1. QM/MM Minimization Steps: 0 2. QM/MM heat up steps: 0 3. QM/MM NPT pressure equilibration steps: 0 4. QM/MM NVE production run steps: 0

User

Yes

Chatbot Job Launched. Job ID: <job-id>

{Button to download output of Step-2}

Do you want to proceed to Step-3?

Please launch a job for Step-3 only after Step-2 Job completes!

(Note: In the preliminary version of AutoSolvateWeb, due to the computational resource limit, the user may need to run this simulation on their own clusters with the generated input files.)

User

Yes

Step - 3

ChatBot

Now

let's start with Microsolvated Cluster Extraction.

This step requires two files: 1) '.prmtop' file 2) '.netcdf' file. These files were created when you ran Step-1 and Step-2. I will load these files for you.

You will have to select the 'ID' of the first frame to be extracted, 'Interval' for cluster extraction and lastly, 'Shell thickness' in Angstrom. Please specify each of them as prompted.

What is the ID of the first frame to extract? The default is 0.

User

50 (To

start extract after 100 ps of equilibration)

ChatBot

What

is the Cluster extraction interval (steps)? Default value is 1.

User

1000

ChatBot Please
provide the Shell thickness in Angstrom. Default value is 4.0.

User 4.0

ChatBot Please
confirm the above inputs to run Step-3.

User Yes

(User confirmed inputs are correct)

ChatBot Job Launched. Job ID: <job-id>
{Button to download output of Step-3}

The AutoSolvateWeb output files are presented in **Supplementary Data 3**.

Text S5. Job Queue

The job queue (**Figure S3a**) uses Celery,⁸ a Python library that provides task queues for handling the execution of jobs. Celery is chosen for its simplicity and fail-safe execution. Redis,⁹ a prominent key-value database is used to store data related to jobs launched by the user. Celery integrates seamlessly with Redis.

Two task queues, “cpuQueue” and “gpuQueue”, handle user-created jobs. The “cpuQueue” is attached to four processes on a 4-core cloud instance, whereas the “gpuQueue” is attached to 1 process on the same cloud instance. However, the user interacts with a single abstracted queue that accepts only two jobs (**Figure S3a**). When the user submits a job, it is queued to either the “cpuQueue” or “gpuQueue” based on its nature: Step-1 and Step-3 jobs are launched on the “cpuQueue”, and Step-2 jobs are launched on the “gpuQueue” due to the requirement for executing TeraChem on GPUs. At any given time, four CPU jobs and one GPU job may run in parallel.

When a user submits a job, Celery creates a unique job ID (**Figure S3b**). Celery stores all job data in the Redis database using this job ID. The job status and output can be retrieved using this unique job ID as a key. Users can also delete a job (**Figure S3c**). However, a job already running may or may not be canceled. Usually, when a user requests the cancellation of a job, the job is marked as “deleted” in the database. When the job reaches the end of the queue, it is ignored. Thus, stopping a job in progress may not be possible. However, the user is safeguarded from this issue. When a user deletes a job, they can immediately add another job to the queue, regardless of the deleted job's status.

Text S6. Installation guide.

- AutoSolvateWeb does not need installation. Users can directly launch the webpage (<https://autosolvate.che230059.projects.jetstream-cloud.org>) in their web browser.
- Upon entering the website, users should click on the “I am human” button and respond to a hCaptcha question to validate that they are human users (**Figure S2**).
- Once the validation is completed, users can click on the chatbot icon in the lower right corner of the webpage (**Figure S2**) and start interacting with the chatbot. A video tutorial is available on YouTube (<https://youtu.be/kBhugQ6cbc0>).

Text S7. Example conversation between the chatbot and the user.

Step - 1

User Open the chat box of AutoSolvateWeb

Chatbot I can automate initial structure generation for explicitly solvated systems. This includes input file preparation. Additionally automated QM/MM trajectory generation and microsolvated cluster extraction is supported for the explicitly solvated systems.

Alright, Let's start by specifying the solute file. You can either:

‘I will upload a .xyz file’

‘Download from PubChem’

User I will upload a .xyz file (user choose upload option)

Chatbot Sure, Please click the button below to upload the solute file

yes

User File Uploaded successfully (sent on user behalf if file uploaded successfully)

ChatBot Please consider providing the name of the solute. If you provide the name of the solute, I can fill most of the parameters required for step-1. Please note that I only support solute compounds that are available on PubChem API. Would you like to provide a solute name?

User Yes (User chooses to provide the solute name)

ChatBot Please enter the IUPAC name of the solute

User naphthalene

ChatBot The solute entered is naphthalene. Please confirm if the solute is correct?

User Yes, it' correct (user confirmed solute name is correct)

ChatBot You have selected naphthalene as solute. The following are the recommended solute parameters for this solute:

Solute Charge: 0

Solute Spin Multiplicity: 1

Solvent Cube Size(Angstrom): 25.21

Do you want to proceed with the above solute parameters?

User Yes

ChatBot Alright, you will have to set the solvent and solvent related parameters manually.

What is the solvent? Currently, I support the following:

Water

Methanol

Chloroform

NMA

Acetonitrile

User Water

ChatBot What charge method do you want to use for force field fitting?

bcc

resp

User bcc

ChatBot All input parameters for Step-1 are validated.

The following are the inputs given for Step -1:

Solvent: water

Charge Method for Force Field Fitting: bcc

Solute Charge is 0

Solute Spin Multiplicity is 1

Solvent Cube Size is 25.21 Angstrom.

Do you want proceed with above inputs for Step -1?

User Yes

ChatBot Job Launched. Job ID:552109a1-82fa-41a6-bffa-ebff6c5c1017

{Button to show output of Step-1}

{Button to download output of Step-1}

Do you want to proceed to Step-2?

Please launch a job for Step-2 only after Step-1 Job completes!

User Yes

Step - 2

ChatBot Let's start with step -2: MD Automation

Would you like to run the run simulation in the dry run mode?

User

No

(user replied 'No', dry run mode will be disabled, parameters are limited)

ChatBot Since you have chosen not run in the 'Dry Run' mode, I have restricted the number of steps to be run in the simulation. The following options will be set to the default values. Classical MD control options:

1. MM minimization steps: 100
2. MM heat up steps: 100
3. MM NPT pressure equilibration steps: 100
4. MM NVE production run steps: 0

Do you want add QM/MM simulation to Step -2 (MD Automation)?

User

Yes

(QM/MM will be added to Step-2)

ChatBot

QM/M

M control options

Please select a QM Method. I support the following methods right now:

{Button to reply 'B3LYP'}

{Button to reply 'HF'}

User

HF

ChatBot The following are the inputs entered for Step-2. Please confirm if these inputs are correct:

Classical MD control options - 1. MM minimization steps: 100 2. MM heat up steps: 100 3. MM NPT pressure equilibration steps: 100 4. MM NVE production run steps: 0

QM/MM control options - 1. QM/MM Minimization Steps: 10 2. QM/MM heat up steps: 10 3. QM/MM NPT pressure equilibration steps: 10 4. QM/MM NVE production run steps: 0

User

Yes

(User confirmed inputs are correct)

Chatbot

Job Launched. Job ID: <job-id>

{Button to download output of Step-2}

Do you want to proceed to Step-3?

Please launch a job for Step-3 only after Step-2 Job completes!

User

Yes

Step - 3

ChatBot

Now

let's start with Microsolvated Cluster Extraction.

This step requires two files: 1) '.prmtop' file 2) '.netcdf' file. These files were created when you ran Step-1 and Step-2. I will load these files for you.

You will have to select the 'ID' of the first frame to be extracted, 'Interval' for cluster extraction and lastly, 'Shell thickness' in Angstrom. Please specify each of them as prompted.

What is the ID of first frame to extract? The default is 0.

User

0

ChatBot

What

is the Cluster extraction interval (steps)? Default value is 1.

User

2

ChatBot

Please

provide the Shell thickness in Angstrom. Default value is 4.0.

User

4.0

ChatBot

Please

confirm the above inputs to run Step-3.

User

Yes

(User confirmed inputs are correct)

ChatBot Job Launched. Job ID:82a0fabb-a684-4747-aafd-15ec202fc910

{Button to download output of Step-3}

The downloaded output files for this demo are provided as **Supplementary Data 1**.

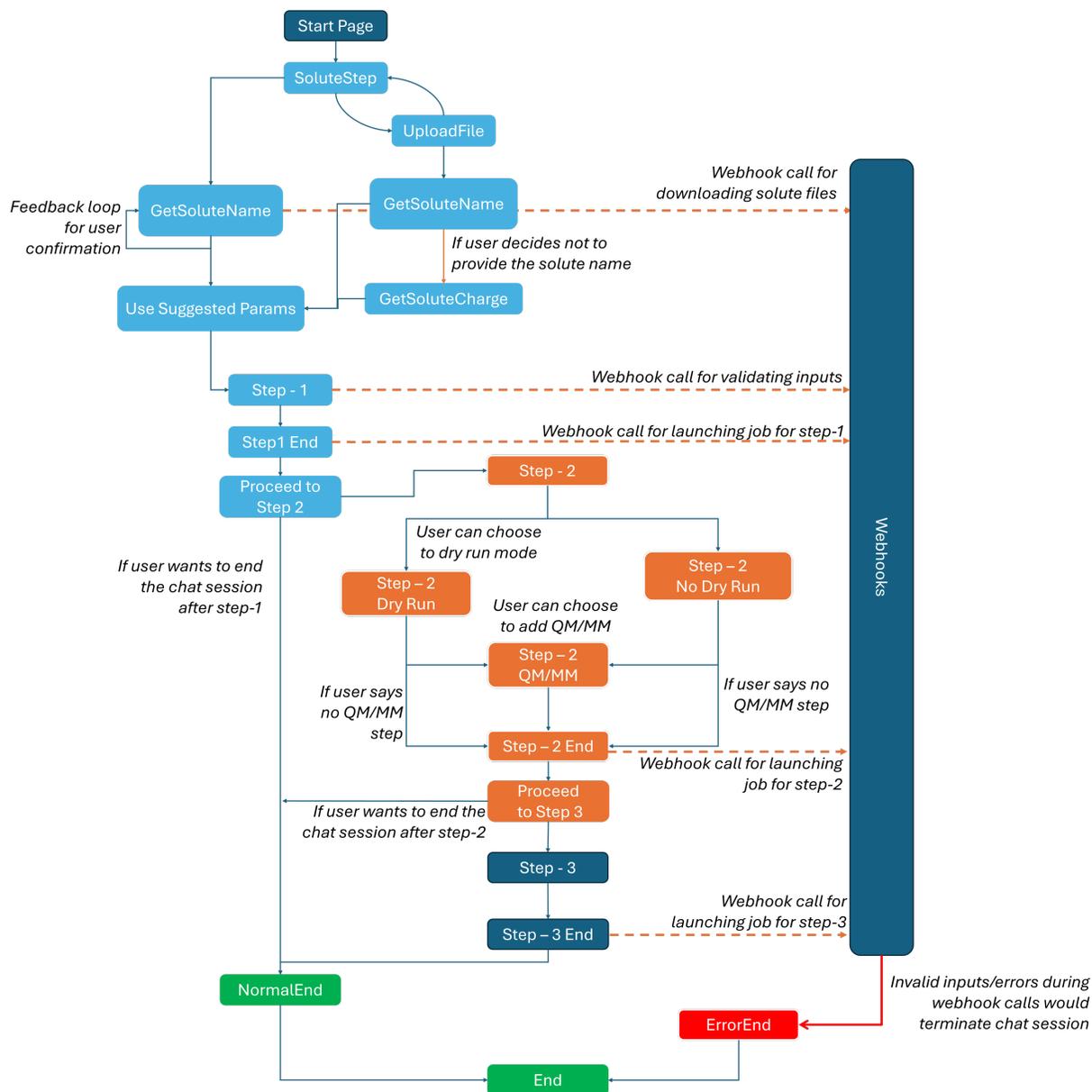


Figure S1. Conversation flow of AutoSolvateWeb chatbot. The conversation with the DialogFlow CX virtual agent can be seen as a state flow diagram, where each state is a page.

AutoSolvate

Please verify to access AutoSolvate

I am human 

Access Autosolvate

This site is protected by hCaptcha and its [Privacy Policy](#) and [Terms of Service](#) apply.

© 2023, Liu Group.

AutoSolvate

This open-source package enables automated initial structure generation for explicitly solvated systems. This includes input file preparation. Additionally automated QM/MM trajectory generation and microsolvated cluster extraction is supported for the explicitly solvated systems.

Current Jobs

Job-Id	Step	Status	Output	Download	Delete
You have no Jobs					



Input

© 2023, Liu Group.

Figure S2. Installation instruction figures.

Figure S3. Job queue figures.

(b) The job queue and the icon to refresh the queue status.

The screenshot shows the AutoSolvate web interface. At the top, there is a navigation bar with links for 'About', 'Source Code', 'Installation Instructions', 'Tutorials', 'Documentation', and 'How to Cite'. The main heading is 'AutoSolvate' with a logo. Below the heading, a brief description of the software is provided. The 'Current Jobs' section features a table with the following data:

Job-Id	Step	Status	Output	Download	Delete
2d1f5735-2920-4855-b21c-abe6066e1189	Step-1	SUCCESS	Show	Download	Delete
d59e7179-82fe-4cf7-b14a-04383545512d	Step-2	PENDING	Show	Download	Delete

Annotations include a red arrow pointing to the 'job queue' label and another red arrow pointing to a refresh icon (a circular arrow) in the top right corner of the table. A text box on the right says 'click this icon to get updated queue status'. Below the table is a progress bar with three steps: 'Step 1 Solvate box and MD parameter generation' (active), 'Step 2 MD automation', and 'Step 3 Microsolvated cluster extraction'. The 'Input' section shows a 3D molecular model and a form with fields for 'Choose File', 'Select Solvent' (set to 'water'), 'Select charge method for force field fitting' (set to 'bcc'), and 'Solute charge'. A 'View' button is next to the 'Choose File' field.

(b) Once a job is launched, its Job ID will be displayed in the chatbox and the job queue.

This screenshot shows the AutoSolvate web interface with a chatbox open. The 'Current Jobs' table now shows the first job as 'PENDING' and the second job as 'SUCCESS'. A red box highlights the Job ID '2d1f5735-2920-4855-b21c-abe6066e1189' in the table. The chatbox, titled 'SolvateBot', contains a message: 'Job Launched. Job ID: 2d1f5735-2920-4855-b21c-abe6066e1189'. A red arrow points from this message to the highlighted Job ID in the table. The chatbox also shows 'Yes' and 'No' buttons, and options to 'Show Output for Step-1' and 'Download Step-1 Output'. The progress bar and input section are also visible.

(c) The job queue only allows two jobs simultaneously. To start the job for Step-3, one should first delete the old job from Step-1.



d initial structure generation for explicitly solvated systems. This includes input file preparation. Additionally automated QM/MM trajectory generation and microsolvated cluster extraction is

Current Jobs

Job-Id	Step	Status	Output	Download	Delete
2d1f5735-2920-4855-b21c-abe6066e1189	Step-1	SUCCESS	Show	Download	Delete
d59e7179-82fe-4cf7-b14a-04383545512d	Step-2	SUCCESS	Show	Download	Delete

click this button to delete a job



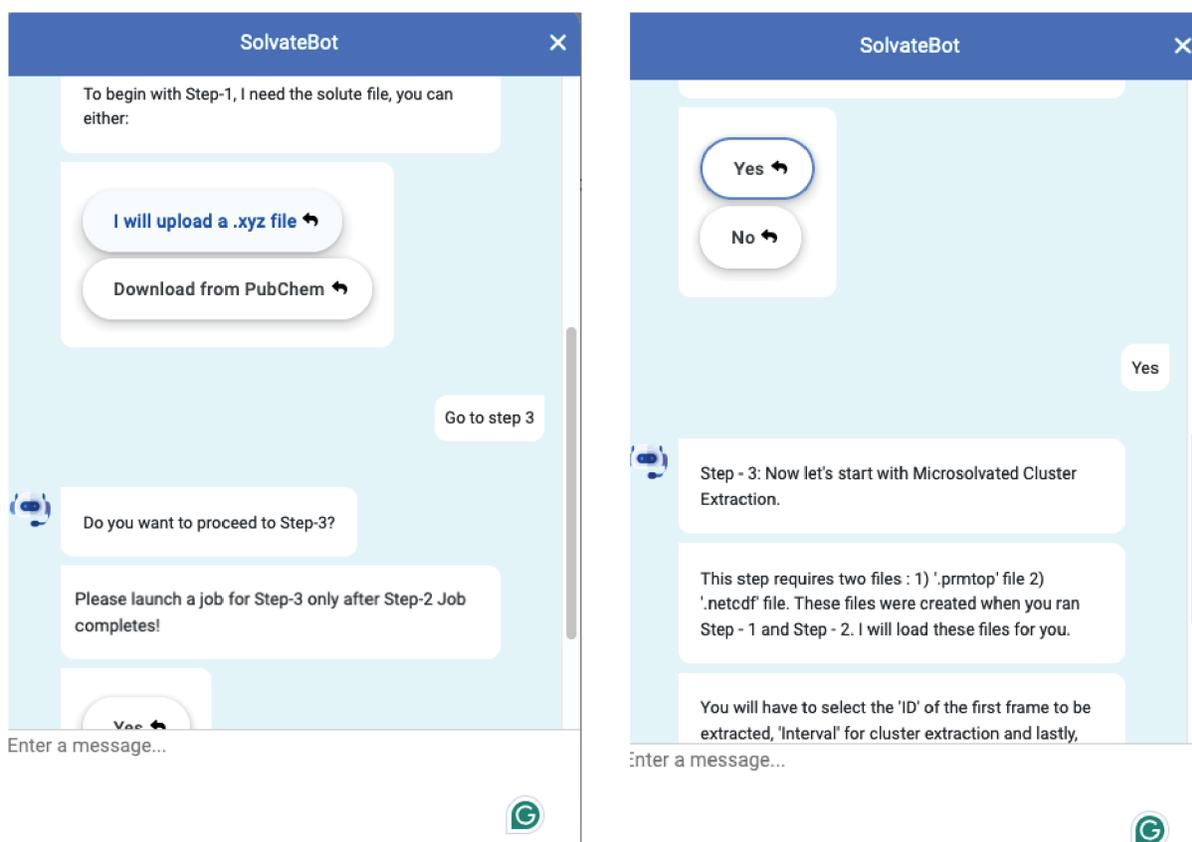


Figure S4. An example conversation containing an intent match for the intent “step3.start”. The beginning of the conversation is on SoluteStep page, where the chatbot asks the user to choose from the two options to specify their solute. However, the user does not respond to the question but send a message that matches the training phrase of the intent “step3.start”. The intent match triggers the route to transition to Step-3, so the chatbot asks the user whether they would like to proceed to Step-3.

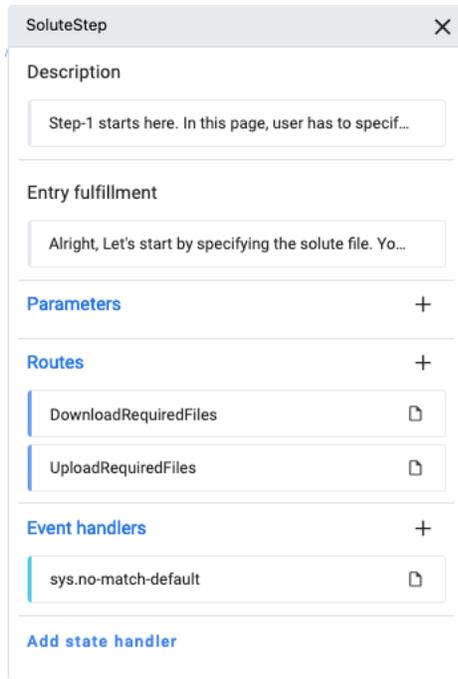


Figure S5. An example page setting with allowed intent routes. The screen snapshot are the settings of the *SoluteStep* page. Allowed intent routes are listed under “Routes” and can be modified by the developer.

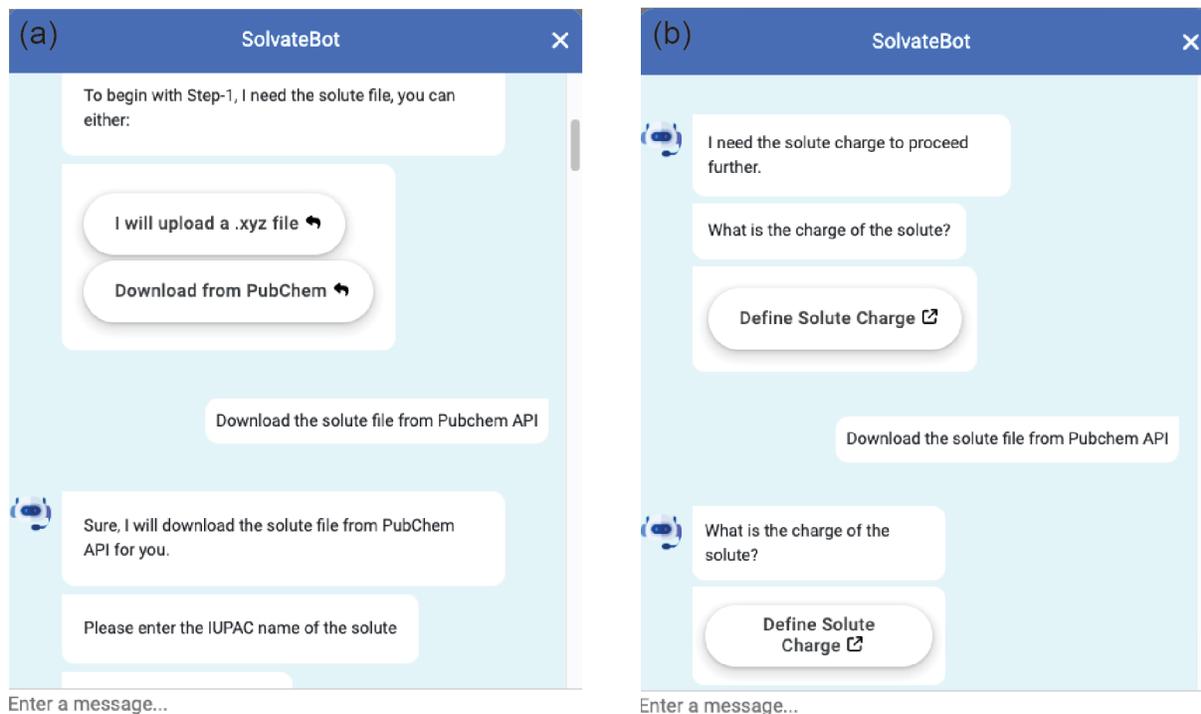


Figure S6. Two example conversations showing that intent match only happens when an intent is in scope. (a) The conversation is on the *SoluteStep* page, where the “DownloadRequiredFiles” is an allowed intent route. Hence, the user’s message “Download the solute file from Pubchem API” triggers an intent match, and the chatbot transition to the *GetSoluteName* page and send messages accordingly. (b) The conversation is on the *GetSoluteName* page, where the intent route “DownloadRequiredFiles” is not in scope. Hence, the user’s message “Download the solute file from Pubchem API” does not lead to an intent match to transition the page. Instead, the chatbot stays on page and repeats the previous question.

Table S1. List of all intents and their training phrases used in AutoSolvateWeb’s chatbot. Highlighted parts in the phrases are annotations, which control how data is extracted from the end-user expression to populate DialogFlow parameters.

Intent	Description	Training phrases
step1.start	Triggers Step-1	“Restart Step-1” “Start the process” “Help me” “I need help with input file preparation.” “Yes. I would like that.”
step2.start	Triggers Step-2	“run step-2” “run step 2 with qm” “MD Automation” “Proceed to Step - 2” “Do MD automation in dry-run mode and run QM/MM step” “MD automation with dry-run mode activated and run QM/MM step” “MD automation with dry-run activated and run QM/MM step” “MD automation in normal mode and run QM/MM step” “Run MD Automation step with normal mode deactivated and run QM/MM step” “Run MD Automation step with normal mode activate and run QM/MM step”
step3.start	Triggers Step-3	“run step-3”; “Run Step -3 with following inputs: id first frame:0 interval:1 step size: 10”; “Do Microsolvated cluster extraction.” “Run Step-3” “Microsolvated cluster extraction”
default welcome intent	Triggers the basic input flow	“yes please”; “repeat once again”; “please start again” “reset everything”; “start again”; “re-run”; “Restart the simulation”; “I would like that.”; “Yes. I need help with”
default negative intent	user wants to end the conversation at any point of time by saying 'GoodBye'	“goodbye”; “No I am Done”; “End the session”; “End the chat”; “No Thanks”; “Good Bye”
UserQuestion	When user asks a question about the parameters	“Define Solvent Cube Size” “What is Solvent cube size?” “What is spin multiplicity?”

		<p>“Define spin multiplicity”</p> <p>“Define charge of the solute”</p> <p>“What is solute charge?”</p> <p>“Define charge method for force field fitting”</p> <p>“Define Solvent”</p> <p>“Define Solute”</p>
DownloadRequiredFiles	Whenever user has to download the following intent will be triggered. Remember any download request should trigger this intent (not just a specific file type like solute file)	<p>“Download the solute file from Pubchem API”</p> <p>“Download the file”</p> <p>“Download from PubChem API”</p>
step1.new	Gives inputs to complete step 1	<p>“2 Angstrom”</p> <p>“solvent cube size is 300”</p> <p>“solute spin multiplicity is 20”</p> <p>“solute charge 1”</p> <p>“My solute charge is 100.”</p> <p>“Use bcc as charge method for force field fitting.”</p> <p>“Use nma as the solvent.”</p> <p>“Select methanol as the solvent and resp as a charge method”</p> <p>“Select bcc charge method for force field fitting”</p> <p>“My solvent is water.”</p>
TaskFailure		<p>“File could not be uploaded”</p> <p>“File upload failed”</p> <p>“The task failed”</p> <p>“File download failed”</p> <p>“File could not be downloaded”</p>
TaskSuccess		<p>“File was uploaded successfully”</p> <p>“File was downloaded successfully”</p>

Table S2. List of all solvents and their force fields in AutoSolvateWeb

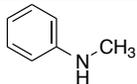
solvent name	IUPAC name	chemical structure	force field
water	oxidane	$\text{H}^{\ominus}\text{O}^{\ominus}\text{H}$	TIP3P ¹⁰
methanol	methanol	$\text{H}_3\text{C}-\text{OH}$	non-polarization force field by P. Kollman <i>et al.</i> ¹¹
chloroform	chloroform		non-polarization force field by P. Kollman <i>et al.</i> ¹¹
NMA	N-Methylaniline		non-polarization force field by P. Kollman <i>et al.</i> ¹¹
acetonitrile	acetonitrile	$\text{N}\equiv\text{C}-\text{CH}_3$	six-site model by C. Jaime <i>et al.</i> ¹²

Table S3. Description of all files in the AutoSolvateWeb output ZIP.

name	file type	usage	remarks
Step-1 generated files			
solute.xyz	XYZ molecular structure file	Storing the element and coordinates of the solute molecule.	
solute.xyz.pdb	Protein data bank (PDB) format structure file	Solute PDB file converted from the solute.xyz file by OpenBabel. Atom names are not uniquely named.	
solute.pdb	Protein data bank (PDB) format structure file	Refined solute PDB file (compared to solute.xyz.pdb) with unique atom names for each atom.	
solute.mol2	Mol2 molecular structure file	Storing the molecular information of the solute, including the element, atom type, coordinates, connectivity, charge, etc.	
solute.frcmod	AMBER force field parameter modification file	Adding the forcefield parameters that are missing from the original GAFF forcefield.	
solute.lib	LEaP library file	A human-readable format generated by the LEaP program. Stores the solute information, including the element, atom type, coordinates, connectivity, charge, and GAFF force field parameters.	
solvated.inpcrd	AMBER format coordinate file	The initial coordinate of the solvated system (solute + solvent box).	

solvated.pdb	Protein data bank format structure file	PDB structure file of the solvated system (solute + solvent box). It can be viewed by many programs, such as Pymol and Avogadro.	
solvated.prmtop	AMBER format parameter/topology file	Storing the charge, topology, and force field parameters for each bond, angle, dihedral, and nonbonded interaction of the whole solvated system.	
resp_scr	folder	This folder contains the temporary output files of the RESP charge fitting of the solute.	Use “resp” for charge fitting
ANTECHAMBER_AC.AC	Antechamber intermediate file	Output information during running Antechamber	
ANTECHAMBER_AC.AC0	Antechamber intermediate file	Output information during running Antechamber	
ANTECHAMBER_AM1BCC_PRE.AC	Antechamber intermediate file	Output information during running Antechamber	Use “bcc” for charge fitting
ANTECHAMBER_AM1BCC.AC	Antechamber intermediate file	Output information during running Antechamber	Use “bcc” for charge fitting
ANTECHAMBER_BOND_TYPE.AC	Antechamber intermediate file	Output information during running Antechamber	
ANTECHAMBER_BOND_TYPE.AC0	Antechamber intermediate file	Output information during running Antechamber	
ATOMTYPE.INF	Antechamber intermediate file	A log file showing how Antechamber determines the atom type for each atom in solute.	
sqm.in	Antechamber intermediate file	Performing a minimization using semi-empirical quantum mechanics methods.	
sqm.out	Antechamber intermediate file	Containing output messages generated during the execution of the sqm.in script.	
sqm.pdb	Protein data bank (PDB) format structure file	Storing the optimized solute molecule, including elements, coordinates, connectivity, and	

		residue name. It can be viewed by many programs, such as Pymol and Avogadro.	
leap_add_solventbox.cmd	LEaP command	Containing commands to add a solvent box around a solute.	
leap_add_solventbox.log	LEaP log	Containing output messages generated during the execution of the leap_add_solventbox.cmd script.	
leap.cmd	LEaP command	Containing commands to generate solute.lib and solute.pdb files based on solute.mol2 and solute.frcmod inputs.	
leap_savelib.log	LEaP log	Containing output messages during the execution of the leap.cmd script.	
leap_add_solventbox.cmd	LEaP command	Containing commands to add a solvent box around a solute.	
leap_add_solventbox.log	LEaP log	Containing output messages generated during the execution of the leap_add_solventbox.cmd script.	
leap.log	LEaP log	Containing output messages generated during the execution of the leap.cmd script.	
Step-2 generated files			
runMM.sh	Shell script	Amber commands to run each step of MM	Dry run mode disabled
runQMMM.sh	Shell script	Amber commands to run each step of QM/MM	Dry run mode disabled
mmmin.in	Amber format input file	Input keywords for a classical mechanics minimization run.	
mmmin.out	Amber format output file	Containing output messages generated during the execution of the mmmin.in script.	Dry run mode disabled
mmmin.info	Amber output file	Information about the energy profile of the system during the MM energy minimization.	Dry run mode disabled
mmmin.netcdf	Network common data form (netcdf)	Containing trajectory file for the classical mechanics minimization	Dry run mode disabled

	format trajectory file	run. Trajectory write-out frequency: 1 frame per 100 MD steps.	
mmheat.in	Amber format input file	Input keywords for a classical mechanics NVT simulation to gradually heat up to a target temperature.	
mmheat.out	Amber format output file	Containing output messages generated during the execution of the mmheat.in script.	Dry run mode disabled
mmheat.info	Amber output file	Information about the energy profile of the system during the heating process.	Dry run mode disabled
solvated-heat.netcdf	Network common data form (netcdf) format trajectory file	Containing a trajectory file for the classical mechanics NVT simulation to gradually heat up to a target temperature. Trajectory write-out frequency: 1 frame per 100 MD steps.	Dry run mode disabled
mmnpt.in	Amber format input file	Input keywords for a classical mechanics NPT simulation to reach a target pressure.	
mmnpt.out	Amber format output file	Containing output messages generated during the execution of the mmnpt.in script.	Dry run mode disabled
mmnpt.info	Amber output file	Information about the energy profile of the system during the NPT simulation.	Dry run mode disabled
solvated-mmnpt.netcdf	Network common data form (netcdf) format trajectory file	Containing a trajectory file for the classical mechanics NPT simulation to reach a target pressure. Trajectory write-out frequency: 1 frame per 100 MD steps.	Dry run mode disabled
mmnve.in	Amber format input file	Input keywords for a classical mechanics NVE simulation.	
mmnve.out	Amber format output file	Containing output messages generated during the execution of the mmnve.in script.	Dry run mode disabled, NVE step > 0
mmnve.info	Amber output file	Information about the energy profile of the system during the NVE simulation.	Dry run mode disabled

			NVE step, > 0
solvated- mmnve.netcdf	Network common data form (netcdf) format trajectory file	Containing trajectory file for the classical mechanics NVE simulation. Trajectory write-out frequency: 1 frame per 100 MD steps.	Dry run mode disabled, NVE step > 0
qmmmin.in	Amber format input file	Input keywords for a QM/MM energy minimization using the Amber/TeraChem interface.	
qmmmin.out	Amber format output file	Containing output messages generated during the execution of the qmmmin.in script.	Dry run mode disabled
qmmmin.info	Amber output file	Information about the energy profile of the system during the MM energy minimization.	Dry run mode disabled
solvated- qmmmin.netcdf	Network common data form (netcdf) format trajectory file	Containing trajectory file for the QM/MM energy minimization. Trajectory write-out frequency: 1 frame per MD step.	Dry run mode disabled
qmmheat.in	Amber format input file	Input keywords for a QM/MM NVT simulation to gradually heat up to a target temperature using the Amber/TeraChem interface..	
qmmheat.out	Amber format output file	Containing output messages generated during the execution of the qmmheat.in script.	Dry run mode disabled
qmmheat.info	Amber output file	Information about the energy profile of the system during the QM/MM heating process.	Dry run mode disabled
solvated- qmmheat.netcdf	Network common data form (netcdf) format trajectory file	Containing a trajectory file for the QM/MM NVT simulation to gradually heat up to a target temperature. Trajectory write-out frequency: 1 frame per MD step	Dry run mode disabled
qmmnvt.in	Amber format input file	Input keywords for a QM/MM NVT simulation at a constant temperature using the Amber/TeraChem interface.	

qmmmnavt.out	Amber format output file	Containing output messages generated during the execution of the qmmmnavt.in script.	Dry run mode disabled
qmmmnavt.info	Amber output file	Information about the energy profile of the system during the QM/MM NVT simulation.	Dry run mode disabled
solvated-qmmmnavt.netcdf	Network common data form (netcdf) format trajectory file	Containing trajectory file for the QM/MM NVT simulation at a constant temperature. Trajectory write-out frequency: 1 frame per MD step	Dry run mode disabled
qmmmnave.in	Amber format input file	Input keywords for a QM/MM NVE simulation using the Amber/TeraChem interface.	Dry run mode disabled, NVE step > 0
qmmmnave.out	Amber format output file	Containing output messages generated during the execution of the qmmmnave.in script.	Dry run mode disabled, NVE step > 0
qmmmnave.info	Amber output file	Information about the energy profile of the system during the QM/MM NVE simulation.	Dry run mode disabled, NVE step > 0
solvated-qmmmnave.netcdf	Network common data form (netcdf) format trajectory file	Containing trajectory file for the QM/MM NVE simulation. Trajectory write-out frequency: 1 frame per MD step	Dry run mode disabled, NVE step > 0
mm.ncrst	Amber format file	Storing coordinates of the atoms in a molecular system for initialize or restart MM simulations	Dry run mode disabled
qmmm.ncrst	Amber format file	Storing coordinates of the atoms in a molecular system for initialize or restart QM/MM simulations	Dry run mode disabled
qmmm_region.pdb	Protein data bank (PDB) format structure file	PDB structure file of the QM region used in the QM/MM simulation	Dry run mode disabled
old.tc_job.dat	TeraChem output	A temporary output file of the gradient evaluation performed by	Dry run mode disabled

		TeraChem during the QM/MM simulation.	
old.tc_job.inp	TeraChem input	A temporary input file of the gradient evaluation performed by TeraChem during the QM/MM simulation.	Dry run mode disabled
inpfile.xyz	Xyz molecular structure file	The input structure for TeraChem gradient calculations performed for QM/MM.	Dry run mode disabled
ptchrg.xyz	Xyz molecular structure file	Point charges of the MM region used for TeraChem gradient calculations performed for QM/MM.	Dry run mode disabled
tc_job.tpl	TeraChem input file template	Containing structure-independent input parameters for running TeraChem by the Amber/TeraChem interface in QM/MM simulations.	Dry run mode disabled
Step-3 generated files			
solvated-cutoutn- <frame>.xyz	XYZ molecular structure file	The structure files of the extracted microclusters. <frame> indicates the frame number in the MM or the QM/MM trajectory. Only elements and coordinates are included.	Can have multiple files with different frames, determined by the user input.

Table S4. Sample Runtimes for selected solutes and solvents.

solute	solvent	charge method	solute charge	spin multiplicity	box size (Å)	heavy atoms	Step-1 time (s)
2-Hydroxybenzoic acid	water	bcc	0	1	24.94	10	17.1
	chloroform	bcc	0	1			15.6
	nma	bcc	0	1			15.7
	acetonitrile	bcc	0	1			19.9
	methanol	bcc	0	1			15.6
	chloroform	resp	1	2	32.94		95.2
	nma	resp	1	2			97.0
	methanol	resp	1	2			96.8
	water	resp	0	1	24.94		50.2
	water	resp	1	2	32.94		97.2
	acetonitrile	resp	1	2			99.7
Ammonia	water	bcc	0	1	19.63	1	2.8
	water	resp	0	1			18.6
	water	resp	1	2			19.1
	chloroform	bcc	0	1	19.63		2.8
	nma	bcc	0	1			2.8
	acetonitrile	bcc	0	1			91.7
	methanol	bcc	0	1			2.9
	chloroform	resp	1	2	27.63		18.7
	nma	resp	1	2			19.0
	acetonitrile	resp	1	2			24.5
	methanol	resp	1	2			19.2
Butanoic acid	water	bcc	0	1	24.60	6	7.8
	chloroform	bcc	0	1			7.9
	nma	bcc	0	1			7.9
	acetonitrile	bcc	0	1			12.4
	methanol	bcc	0	1			8.0
	water	resp	0	1	32.60		25.4
	water	resp	1	2			33.0
	chloroform	resp	1	2			31.4
	nma	resp	1	2			32.4
	acetonitrile	resp	1	2			35.6
	methanol	resp	1	2			32.9
Caffeine	water	bcc	0	1	25.43	14	10.9
	water	resp	0	1			112.4
	water	resp	1	2			246.9
	chloroform	bcc	0	1	25.43		10.9
	nma	bcc	0	1			10.9
	acetonitrile	bcc	0	1			16.0

	methanol	bcc	0	1	33.43		11.0
	chloroform	resp	1	2			244.2
	nma	resp	1	2			246.1
	acetonitrile	resp	1	2			248.1
	methanol	resp	1	2			247.3
Catechin	water	bcc	0	1	29.93	21	34.5
	water	resp	0	1			252.8
	water	resp	1	2	37.93		633.5
	chloroform	bcc	0	1	29.93		34.9
	nma	bcc	0	1			35.3
	acetonitrile	bcc	0	1			38.1
	methanol	bcc	0	1			36.1
	chloroform	resp	1	2	37.93		627.7
	nma	resp	1	2			631.2
	acetonitrile	resp	1	2			632.3
	methanol	resp	1	2			633.8
Dichloromethane	water	bcc	0	1	20.92	3	2.8
	water	resp	0	1			19.5
	water	resp	1	2	28.92		20.5
	chloroform	bcc	0	1	20.92		2.9
	nma	bcc	0	1			2.9
	acetonitrile	bcc	0	1			60.8
	methanol	bcc	0	1			2.9
	chloroform	resp	1	2	28.92		19.9
	nma	resp	1	2			20.5
	acetonitrile	resp	1	2			24.8
	methanol	resp	1	2			21.0
Epigallocatechin gallate	water	bcc	0	1	30.18	33	78.9
	water	resp	0	1			776.2
	water	resp	1	2	38.18		1841.1
	chloroform	bcc	0	1	30.18		78.7
	nma	bcc	0	1			84.0
	acetonitrile	bcc	0	1			80.4
	methanol	bcc	0	1			83.1
	chloroform	resp	1	2	38.18		1839.2
	nma	resp	1	2			1842.2
	acetonitrile	resp	1	2			1840.0
	methanol	resp	1	2			1852.3
Ethanethiol	water	bcc	0	1	22.52	3	2.8
	water	resp	0	1			19.7
	water	resp	1	2	30.52		21.6
	chloroform	bcc	0	1	22.52		2.9
	nma	bcc	0	1			2.9

	acetonitrile	bcc	0	1	30.52		13.2
	methanol	bcc	0	1			2.9
	chloroform	resp	1	2			20.4
	nma	resp	1	2			21.2
	acetonitrile	resp	1	2			25.6
	methanol	resp	1	2			21.9
Porphyrin	water	bcc	0	1	28.45	24	18.6
	water	resp	0	1	36.45		447.9
	water	resp	1	2			922.3
	chloroform	bcc	0	1	28.45		20.6
	nma	bcc	0	1			20.6
	acetonitrile	bcc	0	1			22.9
	methanol	bcc	0	1			21.0
	chloroform	resp	1	2	36.45		917.7
	nma	resp	1	2			920.0
	acetonitrile	resp	1	2			923.2
methanol	resp	1	2	922.3			
Triethyl phosphite	water	bcc	0	1	26.15	10	5.3
	chloroform	bcc	0	1			5.4
	nma	bcc	0	1			5.6
	acetonitrile	bcc	0	1			11.1
	methanol	bcc	0	1			5.7
	water	resp	0	1	34.15		54.6
	water	resp	1	2			79.3
	chloroform	resp	1	2			76.7
	nma	resp	1	2			78.0
	acetonitrile	resp	1	2			80.3
methanol	resp	1	2		79.4		

Table S5. List of labels measured as part of load testing

Label Name	Description
InitialPing	Initial GET request to access the website. The website's HTML, javascript and CSS files are downloaded.
InitialGreeting	API call to initiate handshake with virtual agent
Pubchem	Send 'Download from Pubchem' as user response option to agent
Naphthalene	Send 'Naphthalene' as solute name
soluteConfirmation	Confirm 'Naphthalene' as solute name
confirmParameters	Confirm auto-suggested parameters
Solvent	Select the solvent
ChargeMethod	Select the charge method
Step1Confirm	Confirm all parameters for Step 1 and submit the job.

Table S6. Performance results of each Load Testing

Label	User Count	response time (ms)				Error %
		Avg.	Min.	Max	Std.	
InitialPing	120	340	273	833	71	0
InitialGreeting	120	269	151	480	66	0
Pubchem	120	242	132	376	46	0
Naphthalene	120	244	131	427	55	0
soluteConfirmation	120	822	380	1853	412	0
confirmParameters	120	230	148	399	42	0
Solvent	120	761	173	1109	212	0
ChargeMethod	120	676	139	1087	240	0
Step1Confirm	120	221	134	335	41	0

Table S7. Performance results of each Load Testing – the maximum number of users for the label ‘confirmParameters’

Label	User Count	response time (ms)				Error %
		Avg.	Min.	Max	Std.	
confirmParameters	50	192	126	303	36.74668	0
confirmParameters	120	194	128	288	37.76709	0
confirmParameters	150	234	131	491	69.72391	0
confirmParameters	200	249	97	544	94.63126	0.14

Table S8. Performance results of each Load Testing – maximum number of concurrent users that can load the web interface.

Label	User Count	response time (ms)				Error %
		Avg.	Min.	Max	Std.	
IntialPing	1000	910	374	1684	280.1682	0
IntialPing	9956	7509	648	21075	4186.465	0
IntialPing	25000	10096	5	35004	7581.764	0.4

Table S9. List of all pages and form parameters filled in during the conversation of Autosolvate Chatbot

Page Name	Form Parameters	Routes
Start Page	None	<ul style="list-style-type: none"> • SoluteStep.
SoluteStep	None	<ul style="list-style-type: none"> • Intent-based route to 'UploadFile'. • Intent-based route to 'GetSoluteName'.
UploadFile	None	<ul style="list-style-type: none"> • Intent based route to 'Step-1' if uploaded successfully. • Intent based route to 'SoluteStep' if upload fails.
GetSoluteName	1) Solute Name 2) Confirmation (<i>for Solute Name: Boolean 'Yes' or 'No'</i>)	<ul style="list-style-type: none"> • Condition-based route to 'Use Suggested Params', if the 'Confirmation' parameter is 'Yes'. • Condition-based route to 'GetSoluteName', if the 'Confirmation' parameter is 'No'.
GetSoluteCharge	1) Solute Charge	<ul style="list-style-type: none"> • Condition-based route to fill solute parameters from the uploaded file.
Use Suggested Params	Input Parameters for Step – 1 (<i>e.g., solvent name, solute charge</i>)	<ul style="list-style-type: none"> • Condition-based route to reset all parameters for Step-1.
Step-1	Input Parameters for Step – 1 (<i>e.g., solvent name, solute charge</i>)	<ul style="list-style-type: none"> • Condition based route to 'Step-1 End' if webhook call validates the input parameters as valid. • Condition based route to 'Error End' if webhook call validates the input parameters as invalid.
Step-1 End	None	<ul style="list-style-type: none"> • Condition based route to 'Proceed to Step 2'. Routed from webhook call.
Proceed to Step 2	1) Confirmation (to continue the conversation)	<ul style="list-style-type: none"> • Condition-based route to 'Step-2'. • Condition-based route to 'NormalEnd'.

table continued....

Step-2	<ol style="list-style-type: none"> 1) Temperature (Kelvin) 2) Pressure (Bar) 3) Dry Run Mode 	<ul style="list-style-type: none"> • Condition-based route ‘Step-2 Dry Run’ if Dry Run Mode is True. • Condition-based route ‘Step-2 No Dry Run’ if Dry Run Mode is False.
Step-2 Dry Run	<ol style="list-style-type: none"> 1) Input Parameters for classical MM Step 2) Boolean parameter for QM/MM 	<ul style="list-style-type: none"> • Condition-based route to ‘Step-2 QM/MM’, if the Boolean parameter for ‘QM/MM’ is true. • Condition-based route to ‘Step-2 End’, if the Boolean parameter for ‘QM/MM’ is false.
Step-2 No Dry Run	<ol style="list-style-type: none"> 1) Boolean parameter for QM/MM 	<ul style="list-style-type: none"> • Condition-based route to ‘Step-2 QM/MM’, if the Boolean parameter for ‘QM/MM’ is true. • Condition-based route to ‘Step-2 End’ if the Boolean parameter for ‘QM/MM’ is false.
Step-2 QM/MM	<ol style="list-style-type: none"> 1) Input Parameters for QM/MM Step (<i>User will be prompted only in dry run mode.</i>) 2) Confirmation for all parameters 	<ul style="list-style-type: none"> • Condition-based route based on user confirmation of all parameters.
Step-2 End	None	<ul style="list-style-type: none"> • Condition-based route to ‘Proceed to Step 3’ or ‘ErrorEnd’. Routed from the webhook call.
Proceed to Step 3	<ol style="list-style-type: none"> 1) Confirmation (to continue the conversation) 	<ul style="list-style-type: none"> • Condition-based route to ‘Step-3’. • Condition-based route to ‘NormalEnd’.
Step-3	<ol style="list-style-type: none"> 1) Input Parameters for Cluster Extraction Step. 2) Confirmation for all parameters 	<ul style="list-style-type: none"> • Condition-based route to ‘Step-3 End’, if all parameters are valid and the user confirms. • Condition-based route.
Step-3 End	None	<ul style="list-style-type: none"> • Condition based route to ‘NormalEnd’ or ‘ErrorEnd’. Routed from webhook call.
NormalEnd	None	To conversation ‘END’.
ErrorEnd	None	To conversation ‘END’.
END	None	None.

REFERENCES

1. Google LLC, DialogFlow Documentation, Intents, <https://cloud.google.com/dialogflow/cx/docs/concept/intent>, (accessed 12/22, 2024).
2. Google LLC, DialogFlow Documentation, State Handlers, <https://cloud.google.com/dialogflow/cx/docs/concept/handler>, (accessed 12/22, 2024).
3. C. I. Bayly, P. Cieplak, W. Cornell and P. A. Kollman, A well-behaved electrostatic potential based method using charge restraints for deriving atomic charges: the RESP model, *J. Phys. Chem.*, 1993, **97**, 10269-10280.
4. F. Zahariev, P. Xu, B. M. Westheimer, S. Webb, J. Galvez Vallejo, A. Tiwari, V. Sundriyal, M. Sosonkina, J. Shen, G. Schoendorff, M. Schlinsog, T. Sattasathuchana, K. Ruedenberg, L. B. Roskop, A. P. Rendell, D. Poole, P. Piecuch, B. Q. Pham, V. Mironov, J. Mato, S. Leonard, S. S. Leang, J. Ivanic, J. Hayes, T. Harville, K. Gururangan, E. Guidez, I. S. Gerasimov, C. Friedl, K. N. Ferreras, G. Elliott, D. Datta, D. D. A. Cruz, L. Carrington, C. Bertoni, G. M. J. Barca, M. Alkan and M. S. Gordon, The General Atomic and Molecular Electronic Structure System (GAMESS): Novel Methods on Novel Architectures, *Journal of Chemical Theory and Computation*, 2023, **19**, 7031-7055.
5. R. Cammi and B. Mennucci, Linear response theory for the polarizable continuum model, *The Journal of Chemical Physics*, 1999, **110**, 9877-9886.
6. R. Improta, V. Barone, G. Scalmani and M. J. Frisch, A state-specific polarizable continuum model time dependent density functional theory method for excited state calculations in solution, *The Journal of Chemical Physics*, 2006, **125**, 054103.
7. S. Hirata and M. Head-Gordon, Time-dependent density functional theory within the Tamm–Dancoff approximation, *Chemical Physics Letters*, 1999, **314**, 291-299.
8. A. Solem, Celery - Distributed Task Queue. *Journal*.
9. S. Sanfilippo, Redis - The Real-time Data Platform. *Journal*.
10. W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey and M. L. Klein, Comparison of simple potential functions for simulating liquid water, *J. Chem. Phys.*, 1983, **79**, 926-935.
11. P. Cieplak, J. Caldwell and P. Kollman, Molecular mechanical models for organic and biological systems going beyond the atom centered two body additive approximation: aqueous solution free energies of methanol and N - methyl acetamide, nucleic acid base, and amide hydrogen bonding and chloroform, *J. Comput. Chem.*, 2001, **22**, 1048-1057.
12. X. Grabuleda, C. Jaime and P. A. Kollman, Molecular dynamics simulation studies of liquid acetonitrile: New six-site model, *J. Comput. Chem.*, 2000, **21**, 901-908.