

User Guide

© 2025 Computational Biology Laboratory,
The University of Tokyo
<https://www.bi.a.u-tokyo.ac.jp/>
All Rights Reserved.

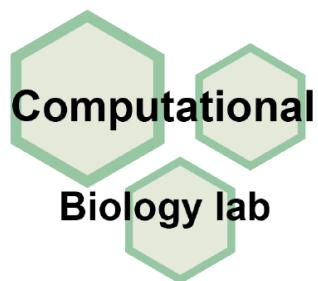


Table of Contents

1. Introduction
2. How to Obtain a Hugging Face Access Token
3. Preparing Input Files
4. Running ColabReaction
5. Analyzing the Calculation Results
6. Installation on a Local Environment
7. Citations and References

Introduction

ColabReaction is a web-based computational chemistry tool designed for rapid and intuitive analysis of molecular reaction pathways.

It runs on Google Colaboratory, eliminating the need for complex installation procedures — you can get started right away using only your web browser.

- ⌚ Visualize reaction pathways in just a few minutes
- ⌚ Obtain high-accuracy results without complex command-line operations
- ✍️ Designed for both beginners and advanced researchers

You can access the tool from the official website below:

🔗 <https://ColabReaction.net>

🔧 Source Code and Technical Details

The source code for ColabReaction is publicly available on GitHub:

💻 <https://github.com/BILAB/ColabReaction>

You can explore, modify, and extend the internal logic of the notebook for your own research needs.

❓ Questions and Feedback

If you have any questions, encounter bugs, or would like to suggest new features, please use the “Issues” page on the GitHub repository to contact the development team:

➡️ [GitHub Issues Page](#)

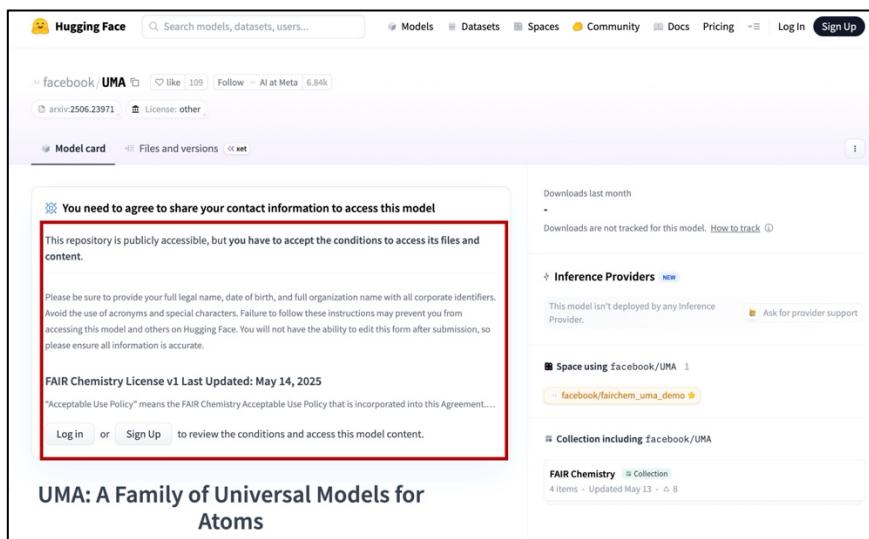
Your feedback is greatly appreciated and will help improve future versions of ColabReaction.

How to Obtain a Hugging Face Access Token

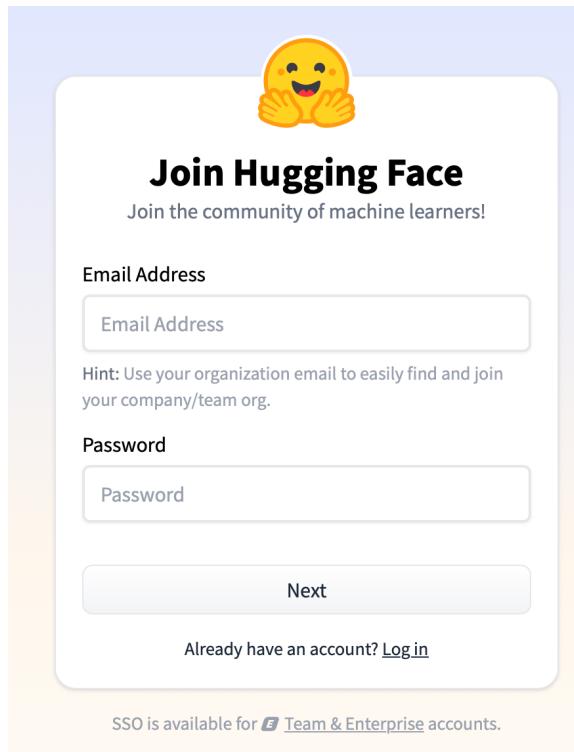
1. Before using ColabReaction, please visit Hugging Face website.

<https://huggingface.co/facebook/UMA> and request access to the UMA model.

You will need a Hugging Face account to proceed. Click on the "Sign Up" button (highlighted in red in the figure below) to create an account.



2. Set your email address and password.



3. Choose a username for your account.

The screenshot shows a 'Complete your profile' form. It includes fields for 'Username' and 'Full name'. There are optional sections for 'Avatar' (with an 'Upload file' button), 'Twitter username' (with a 'Twitter account' link), 'GitHub username' (with a 'GitHub username' link), and 'LinkedIn profile' (with a 'LinkedIn profile' link). There is also a 'Homepage' field and an 'AI & ML interests' field. At the bottom, there is a checkbox for accepting the 'Terms of Service' and 'Code of Conduct', followed by a 'Create Account' button.

4. You will receive a confirmation email from Hugging Face. Click the link provided in the message that says: “Confirm your email address by clicking on this link:”
5. Once your account is created, you will see a page like the one below. Click “Expand to review and access”.

The screenshot shows a Hugging Face model card for 'facebook/UMA'. It includes a 'Model card' tab, 'Files and versions', and 'Inference Providers' sections. A red box highlights the 'Model card' section, which contains a warning: 'You need to agree to share your contact information to access this model'. Below this, it says 'This repository is publicly accessible, but you have to accept the conditions to access its files and content.' and provides instructions for providing full legal name, date of birth, and organization name. At the bottom of this section is a 'Expand to review and access' button. The rest of the page shows repository details like license ('arxiv:2506.23971'), files ('License: other'), and inference providers ('None').

6. Fill in the required fields in the form at the bottom of the page. Check the box “I accept the terms and conditions”, then click “Agree and send request to access repo.”

By agreeing you accept to share your contact information (email and username) with the repository authors.

First Name

Last Name

Date of birth

Country

Affiliation

I accept the terms and conditions
Your country and region (based on approximate Internet address) will be shared with the model owner.

Agree and send request to access repo

7. After submitting the request, you will see a confirmation page. It may take a short while for your access to be approved.

 **You need to agree to share your contact information to access this model**

This repository is publicly accessible, but **you have to accept the conditions to access its files and content.**

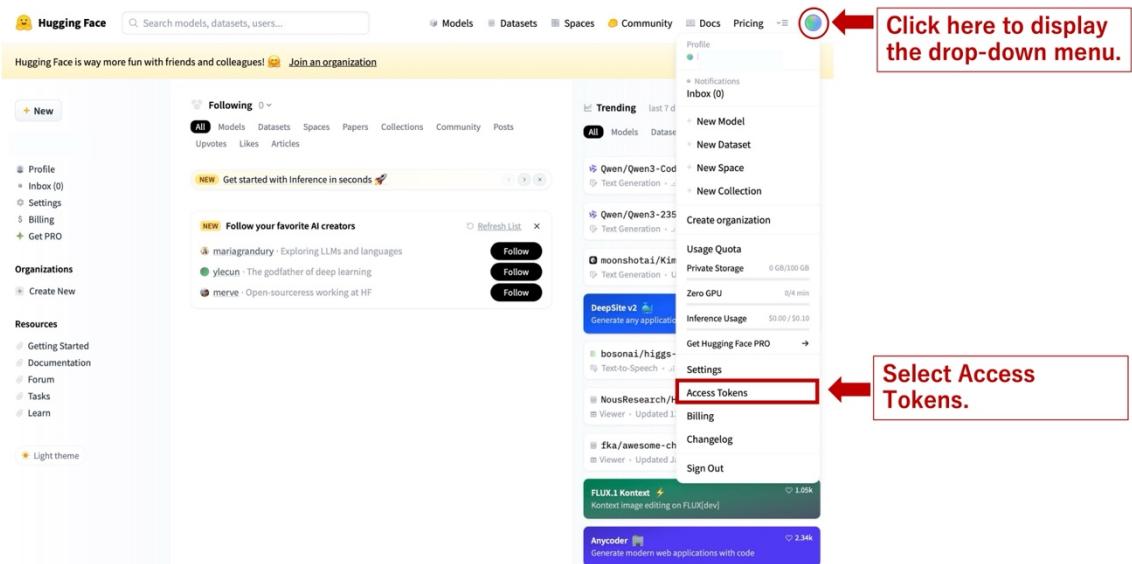
Please be sure to provide your full legal name, date of birth, and full organization name with all corporate identifiers. Avoid the use of acronyms and special characters. Failure to follow these instructions may prevent you from accessing this model and others on Hugging Face. You will not have the ability to edit this form after submission, so please ensure all information is accurate.

FAIR Chemistry License v1 Last Updated: May 14, 2025

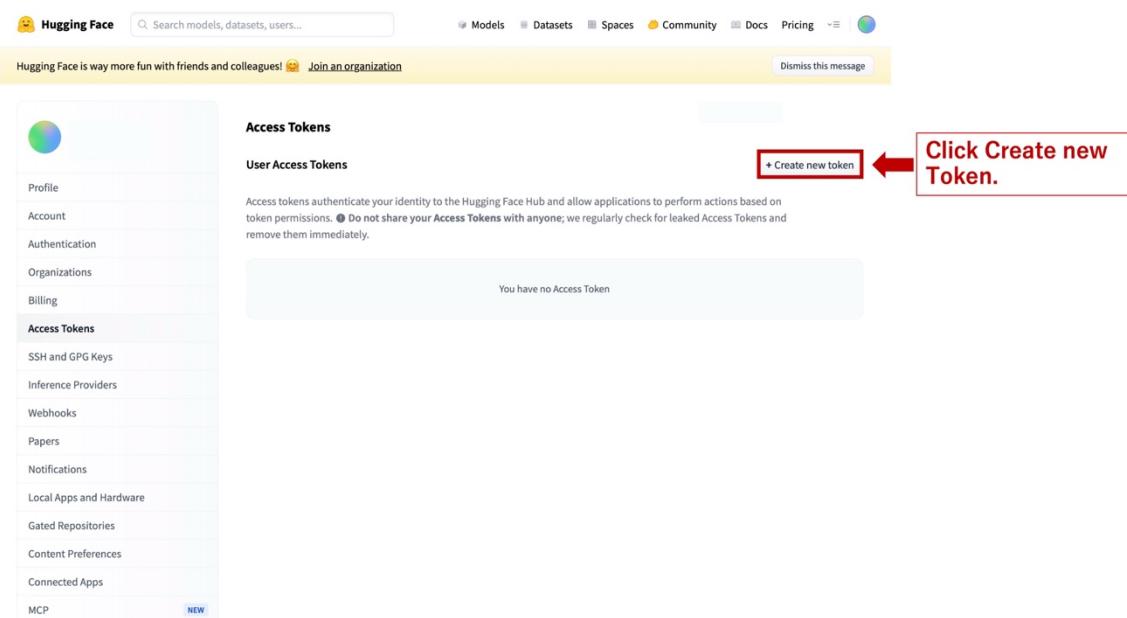
“Acceptable Use Policy” means the FAIR Chemistry Acceptable Use Policy that is incorporated into this Agreement....

Your request to access this repository has been submitted and is awaiting a review from the repository authors. You can check the status of all your access requests in [your settings](#).

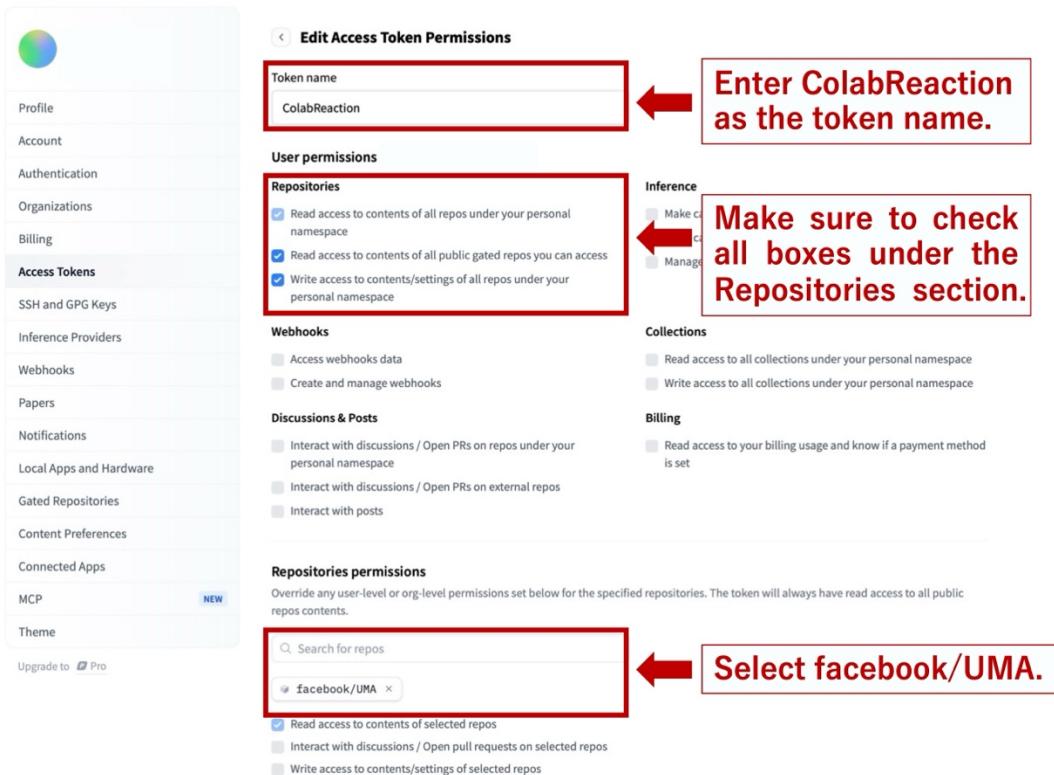
8. Once your request is approved, click “Access Tokens” from the user menu in the top right corner.



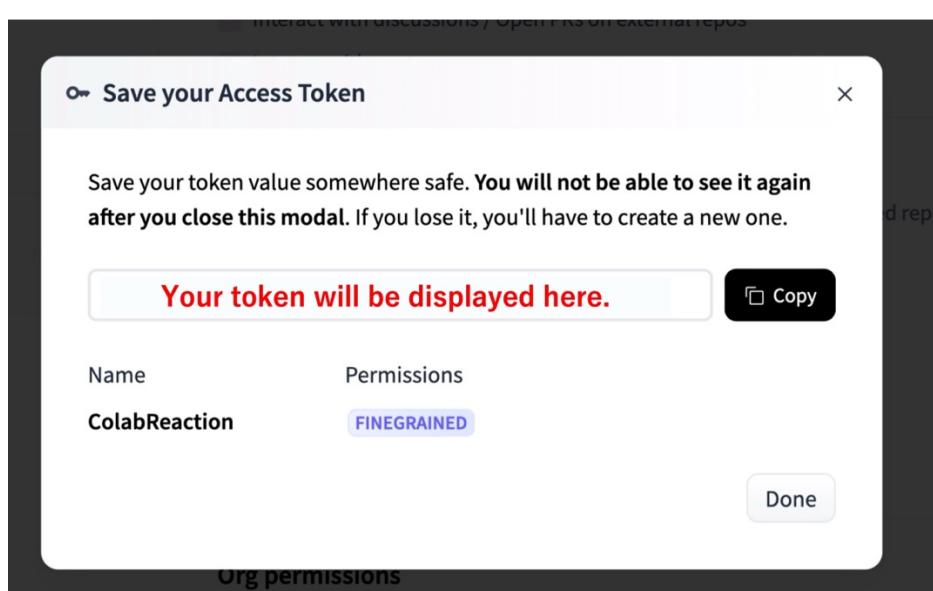
9. Click “+ Create new token” in the top right. Under Repositories permissions, select facebook/UMA, and create a new token.



10. For the Token name, we recommend entering “ColabReaction”, though any name is acceptable. Make sure to check all boxes under the Repositories section. Be sure to select facebook/UMA in the Repositories permissions section.



11. The Token Value will be displayed — be sure to save it, as it will be required for future steps in ColabReaction.



Preparing Input Files

Creating Molecules Using Molecular Visualization Software

In ColabReaction, molecular structure files for the reactant and product are required as inputs.

These molecular structures can be created using standard molecular editors such as:

- [GaussView](#) (commercial molecular editor from Gaussian)
- [Avogadro](#) (open-source molecular editor)

Once the structures are prepared, please save them in one of the following supported formats:

.xyz, .sdf, .mol, .com, .gjf, and .pdb.

⚠ Important: Atom Numbering Consistency

ColabReaction uses a double-ended method, which estimates reaction pathways based on the structures of the reactant and product.

This method assumes that the atom order (numbering) is exactly the same in both structures.

❗ Example:

If Atom 1 is C, Atom 2 is O, and Atom 3 is H in the reactant,
then Atom 1, 2, and 3 must also be C, O, and H respectively in the product.

If the atom order does not match, interpolation along the reaction coordinate or structural alignment will fail, which may lead to incorrect or unrealistic pathways. We strongly recommend verifying and adjusting atom order manually using your molecular editor after building the structures.

In GaussView, you can adjust atom order manually via:

Tools > Atom List

If the atom order is inconsistent, you may observe abnormal behavior in the animation — such as atoms flying apart or reacting unnaturally.

Running ColabReaction

ColabReaction is a notebook-based application that runs on Google Colaboratory.

Please follow the steps below to operate the tool effectively.

① Accessing ColabReaction

Begin by accessing ColabReaction via the following link:

🔗 <https://ColabReaction.net>

② Notebook Structure and Execution

The ColabReaction notebook is divided into two main sections:

- Setup Section: For uploading input files and setting calculation parameters
- Execution Section: For running the actual reaction path calculations

In Google Colaboratory, click the button at the top-left of each code cell to run it. You can also select a cell and press Shift + Enter to run the cell.

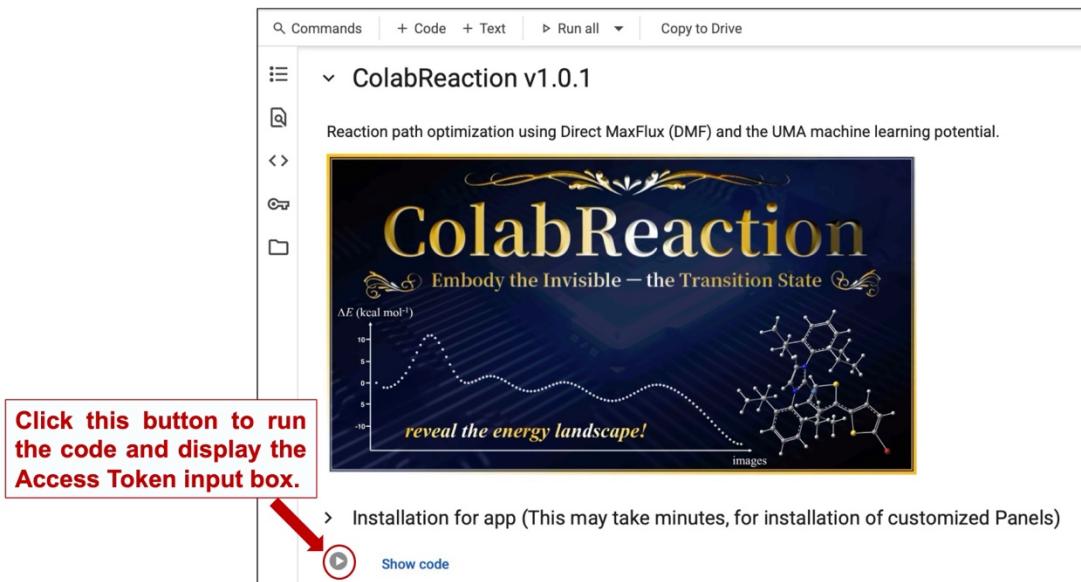
③ Setup Section

This section contains four steps. Simply run each cell in order to complete the initial setup.

Installation for App

Installs all required dependencies (e.g., ASE, fairchem).

⌚ Estimated time: - 2 minutes



⚠ Note: You may see a warning related to the version of the Panel library. This can be safely ignored — it does not affect the performance or results of ColabReaction.

```
Attempting uninstall: panel
Found existing installation: panel 1.7.4
Uninstalling panel-1.7.4:
Successfully uninstalled panel-1.7.4
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
holoviews 1.21.0 requires panel>=1.0, but you have panel 0.0.post1.dev4642+gd80cae which is incompatible.
Successfully installed comm-0.0.2.2 ipywidgets-8.1.7 jedi-0.19.2 jupyter_bokeh-4.0.5 panel-0.0.post1.dev4642+gd80cae panel-3dmol-0.1.0 plotly-6.2.0 py3dmol-2.5.1 rdkit-2025.3.3 widgetsnbextension-4.0.14
real    2m12.162s
user   1m31.910s
sys    0m12.615s
```

Step 1 – Uploading Structure Files

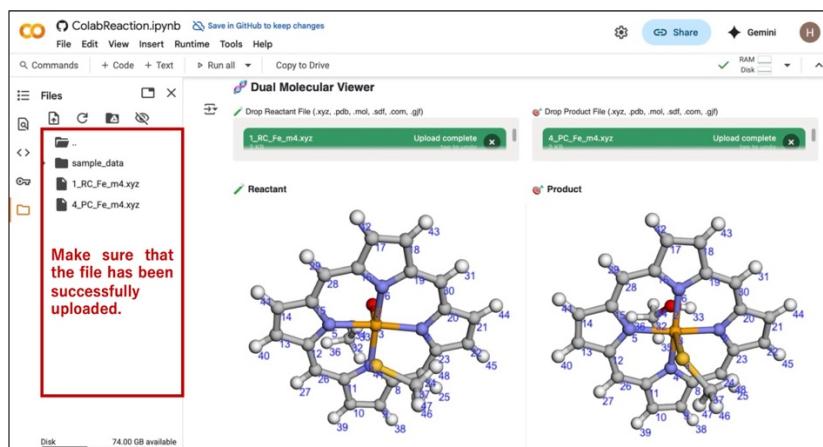
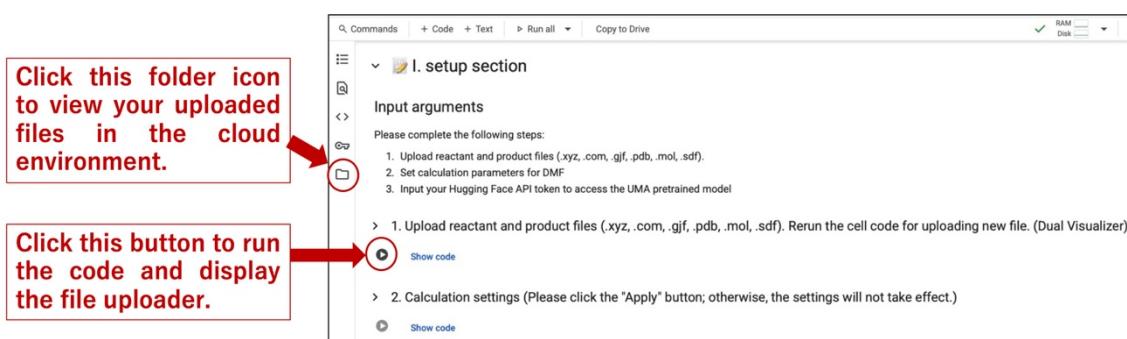
Click to display the uploader.

Estimated time: ~30 seconds

Upload your reactant and product files in any of the following supported formats:
.xyz, .com, .gjf, .pdb, .mol, .sdf

Once uploaded, the 3D structures will be visualized in the notebook.

Be sure to check the file list in the left sidebar of Google Colaboratory to confirm that your files were successfully uploaded to the cloud environment.



Step 2 – Setting Calculation Parameters

Enter the parameters required for the calculation, such as:

- Molecular charge
- Spin multiplicity
- Number of path steps, etc.

➡ After entering the parameters, be sure to click the “Apply” button. Your settings will not take effect unless you do so.

At the start of Step 2, the input files are checked. If there are inconsistencies in molecular composition or atom numbering between the reactant and product, an error message will be displayed.

> 2. Calculation settings (Please click the "Apply" button; otherwise, the settings will not take effect.)

The screenshot shows a Jupyter Notebook cell with a play button icon and a 'Show code' link. Below it is a 'Calculation Settings' panel with the following controls:

- Charge: 0
- Mult: 1
- nmove: 20
- update_teval (unchecked)
- Convergence dropdown set to 'tight'

A text box displays the current settings: 'Current: Charge=0, Mult=1, nmove=20, update_teval=False, Convergence=tight'. At the bottom is a red-bordered 'Apply' button with a checkmark icon.

Be sure to click the Apply button at the end.

Step 3 – Entering Your Hugging Face Token

To use the UMA model, enter the Hugging Face Access Token you previously obtained.

When you run this cell, a field will appear to input your Hugging Face Token. Paste the token and press Enter to proceed.

The screenshot shows a Jupyter Notebook cell with a play button icon and a 'Show code' link. A red box highlights the play button with the text: 'Click this button to run the code and display the Access Token input box.' An arrow points from this text to the play button. Below the cell is a text input field with placeholder text: '... Enter your Hugging Face access token. The input will be hidden for security.' A red box highlights the input field. To its right, red text reads: 'Hugging Face token: [REDACTED] After entering your Access Token, press the Enter key to continue.'

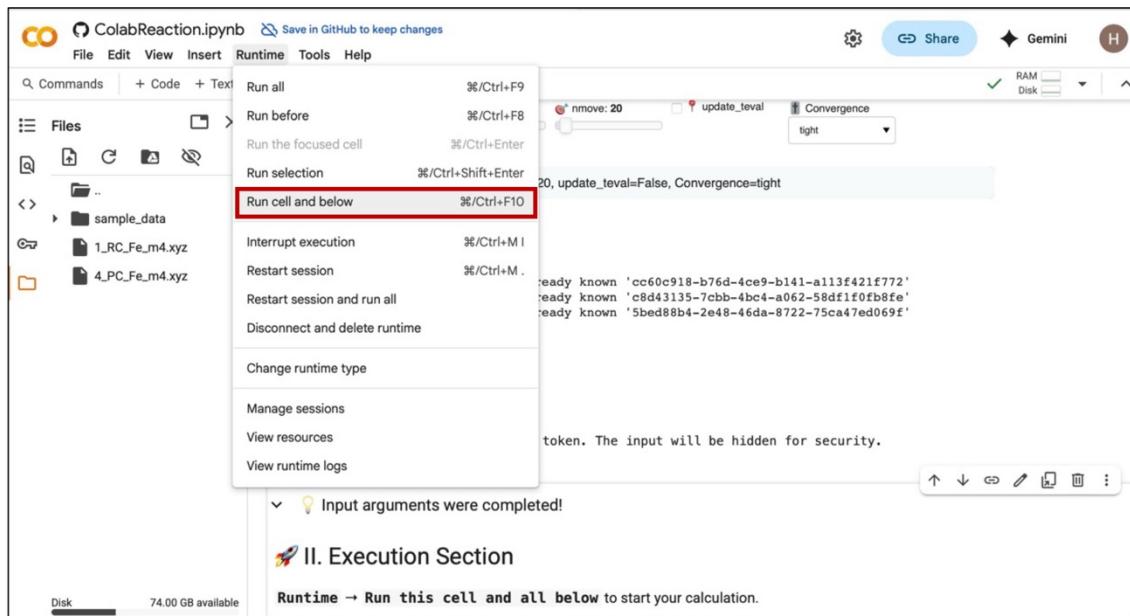
④ Execution Section

Once the setup is complete, proceed to the "II. Execution Section".

1. Click to select the II. Execution Section cell.
2. From the top menu, go to "Runtime" → "Run cell and below".

⌚ During execution, cells will be marked as "Running...".

Depending on the molecular size and conditions, the calculations may take anywhere from a few minutes to about 20 minutes.



⑤ Retrieving the Output

Once all calculations are complete, a .zip archive containing all output files will be downloaded automatically.

🔑 Key Output Files

DMF_final

- The final reaction pathway optimized by DMF/UMA.
- .xyz: Atomic coordinates in XYZ format.
- .traj: ASE trajectory file.
- .pdb or .sdf: Generated when input files are in .pdb, .sdf, or .mol2 format.
- _gv.log: GaussView-compatible file for reaction pathway animation.

DMF_energy.csv

- Energies along the optimized reaction path by DMF/UMA.
- “Image” refers to the energy evaluation points.

DMF_ipopt.out

- Optimization log output by IPOPT during the DMF calculation.
-

Supplementary Files

DMF_tmax

- Structures at the maximum energy point (tmax) along the path in each iteration.
- .xyz: XYZ format.
- .traj: ASE trajectory file.

timing.log

- Log of the calculation time.

DMF_init

- Initial path generated using Correlated Flat-Bottom Elastic Network Model (CFB-ENM).
- .xyz: XYZ format.
- .traj: ASE trajectory file.

fbenm_ipopt.out

- Optimization log from IPOPT during the CFB-ENM initial path construction.

energy_history.txt

- Energies at each iteration during optimization.

force_history.txt

- Atomic forces at each iteration during optimization.

local_maxima

- .log: Vibration result file compatible with GaussView. Users can view the vibration animation in GaussView via Results > Vibrations.
- .txt: Text file containing vibration calculation results.

Analyzing the Calculation Results

1. Visualizing the Reaction Path Animation

After the calculation is complete, the “Animated Molecular Reaction Path” interface will appear.

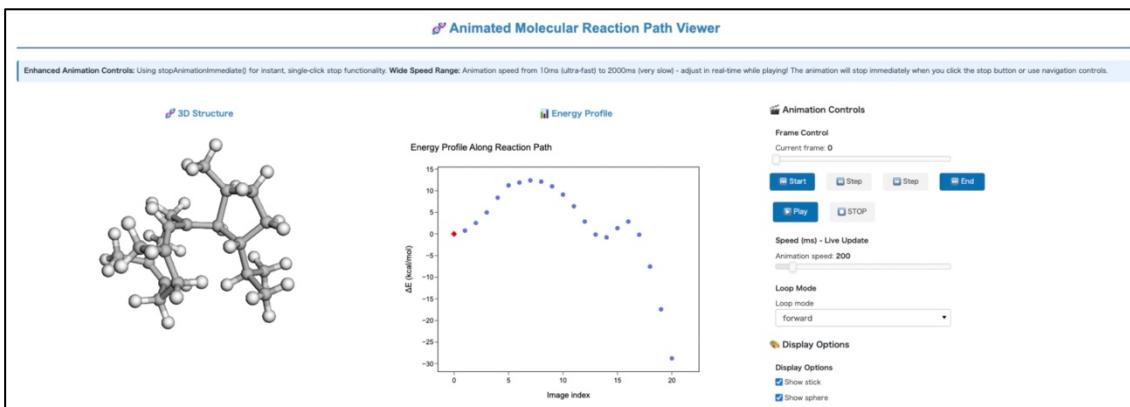
- On the left, the 3D molecular structure is displayed.
- In the center, an energy diagram shows the reaction profile.
- On the right, animation playback controls are provided.

Clicking any point on the energy diagram displays the corresponding 3D molecular structure.

To play the animation, click the Play button on the controller.

You can adjust the playback speed using the Speed slider in the center of the controller.

To step through the reaction one frame at a time, use the Step button.



2. Notes on Vibrational Analysis and Transition State Structures

ColabReaction performs vibrational frequency analysis at each local maximum along the energy diagram.

It then displays the top 3 imaginary (negative) vibrational modes as animations.

Please note:

- The vibrational analysis using UMA/ASE is approximate, not rigorous.
- As a result, multiple imaginary frequencies may appear.
- The absolute values and ranking of the vibrational modes may not be accurate.

You should examine the vibrational modes and select the one that clearly corresponds to a reaction coordinate.

Use that structure as the starting point for a transition state (TS) optimization calculation.

⚠ Important Note

The structures identified as local maxima by ColabReaction are often very close to true transition states, but they are not guaranteed to be exact transition state geometries.

Similarly, the energy diagram is a qualitative representation of the reaction pathway and does not provide quantitatively reliable energies.

If you intend to publish your results, we strongly recommend performing the following:

- Transition state optimization starting from the identified energy-maximum structure
- A subsequent IRC (Intrinsic Reaction Coordinate) calculation

These steps are essential to rigorously identify the true transition state and to validate the reaction pathway.

Installation on a Local Environment

Although we recommend running this notebook on Google Colaboratory, it can also be installed and executed in a local computing environment.

The procedure described below was tested on a standalone local server running Fedora 40 (conda 25.7.0) and on a PC cluster managed by the Torque job scheduler running Rocky Linux 8.9 (conda 25.3.1).

The installation was performed using Miniconda3 for environment management. The following step-by-step instructions provide a reproducible setup for the software environment.

1. Create and activate a conda environment

```
conda create -n colabreaction python=3.11 -y  
conda activate colabreaction
```

2. Configure conda channels

```
conda config --env --remove-key channels 2>/dev/null || true  
conda config --env --add channels pytorch  
conda config --env --add channels nvidia  
conda config --env --add channels conda-forge  
conda config --env --add channels defaults  
conda config --env --set channel_priority strict
```

3. Install essential packages

```
conda install -y -c conda-forge python=3.11 nodejs=20 rdkit numba ¥  
jupyterlab ipykernel  
python -m pip install --upgrade pip
```

4. Install PyTorch with CUDA support

```
python -m pip install --index-url https://download.pytorch.org/whl/cu124 ¥  
torch==2.6.0 torchvision==0.21.0 torchaudio==2.6.0
```

5. Install additional dependencies

```
python -m pip install --no-cache-dir "panel @ ¥  
git+https://github.com/luvwinnie/panel@d80acae43fa3"  
python -m pip install --no-cache-dir "panel-3dmol @ ¥  
git+https://github.com/luvwinnie/panel-3dmol.git@1a4398b66c2a"
```

```
conda install -y -c conda-forge numpy scipy ase cyipopt
```

```
python -m pip install --no-cache-dir "direct-maxflux @  
git+https://github.com/shin1koda/dmf.git@e0edf505136467fd5af8e3ba820e0  
e61d3b782ae"
```

6. Install PyTorch Geometric libraries

```
python -m pip install --no-cache-dir pyg-lib torch-scatter torch-sparse torch-  
cluster torch-spline-conv -f https://data.pyg.org/whl/torch-2.6.0+cu124.html  
python -m pip install --no-cache-dir torch-geometric
```

7. Install visualization and auxiliary packages

```
python -m pip install --no-cache-dir ¥  
"param==2.2.1" "jupyter_bokeh==4.0.5" "comm==0.2.3" "plotly==6.3.0"  
"py3Dmol==2.5.2" ¥  
"fairchem-core==2.3.0" ¥  
"direct-maxflux @  
git+https://github.com/shin1koda/dmf.git@e0edf505136467fd5af8e3ba820e0  
e61d3b782ae"
```

```
conda install -y -c conda-forge ipywidgets
```

The dmf repository was renamed to **pydmf** in December 2025.

To ensure reproducibility and compatibility with existing codes (import dmf), we install dmf from a fixed Git commit prior to the renaming.

8. Register the kernel

```
python -m ipykernel install --user --name=colabreaction --display-name ¥  
"Python (colabreaction)"
```

9. Launching JupyterLab

Before launching JupyterLab, please ensure that the file **Local_ColabReaction.ipynb** is placed in the working directory from which you will start JupyterLab. This ensures that all relative paths and dependencies are correctly resolved.

9-1. On a standalone local server

(run on the server):

```
jupyter lab --no-browser --port=8888 --ip=127.0.0.1
```

(run on your local machine, in a web browser):

Open the following URL:

<http://localhost:8888>

9-2. Within a Torque-managed environment

(run on the server, to allocate a compute node):

```
qsub -l -q <queue_name> -l ¥
```

```
nodes=<compute_node>:ppn=8:gpus=1,walltime=01:00:00
```

```
jupyter lab --no-browser --port=8888 --ip=127.0.0.1
```

(run on the server, once the compute node is allocated):

```
ssh -J <username>@<headnode> -L 8890:localhost:8888
```

<username>@<compute_node>

(run on your local machine, from a separate terminal):

Open the following URL:

<http://localhost:8890>

Note: For convenience, it is recommended to configure the LocalForward option in your `~/.ssh/config` file.

Citations and References

If you use ColabReaction in your research, please cite the following publication:

1. Karasawa, M.; Leow, C. S.; Yajima, H.; Arai, S.; Nishizaki, H.; Terada, T.; Sato H. *ChemRxiv* **2025**. DOI: [10.26434/chemrxiv-2025-zvkqk] (<https://dx.doi.org/10.26434/chemrxiv-2025-zvkqk>)

We also recommend citing the following references related to the underlying DMF/UMA methodology:

2. Nakano, M.; Karasawa, M.; Ohmura, T.; Terada, T.; Sato, H. *ChemRxiv* **2025**. DOI: [10.26434/chemrxiv-2025-md8k6-v2] (<https://dx.doi.org/10.26434/chemrxiv-2025-md8k6-v2>)
3. Koda, S.; Saito, S. Locating Transition States by Variational Reaction Path Optimization with an Energy-Derivative-Free Objective Function. *J. Chem. Theory Comput.* **2024**, 20 (7), 2798-2811.
4. Koda, S.; Saito, S. Flat-Bottom Elastic Network Model for Generating Improved Plausible Reaction Paths. *J. Chem. Theory Comput.* **2024**, 20 (16), 7176-7187.
5. Koda, S.; Saito, S. Correlated Flat-Bottom Elastic Network Model for Improved Bond Rearrangement in Reaction Paths. *J. Chem. Theory Comput.* **2025**, 21 (7), 3513-3522.
6. Wood, B. M.; Dzamba, M.; Fu, X.; Gao, M.; Shuaibi, M.; Barroso-Luque, L.; Abdelmaqsoud, K.; Gharakhanyan, V.; Kitchin, J. R.; Levine, D. S.; et al. UMA: A Family of Universal Models for Atoms. *arXiv preprint* **2025**, <https://ai.meta.com/research/publications/uma-a-family-of-universal-models-for-atoms>.
7. Levine, D. S.; Shuaibi, M.; Spotte-Smith, E. W. C.; Taylor, M. G.; Hasyim, M. R.; Michel, K.; Batatia, I.; Csányi, G.; Dzamba, M.; Eastman, P.; et al. The Open Molecules 2025 (OMol25) Dataset, Evaluations, and Models. *arXiv preprint* **2025**, arXiv:2505.08762. [physics.chem-ph]
8. fairchem; <https://github.com/facebookresearch/fairchem>