

A New Computational Interface for Catalysis

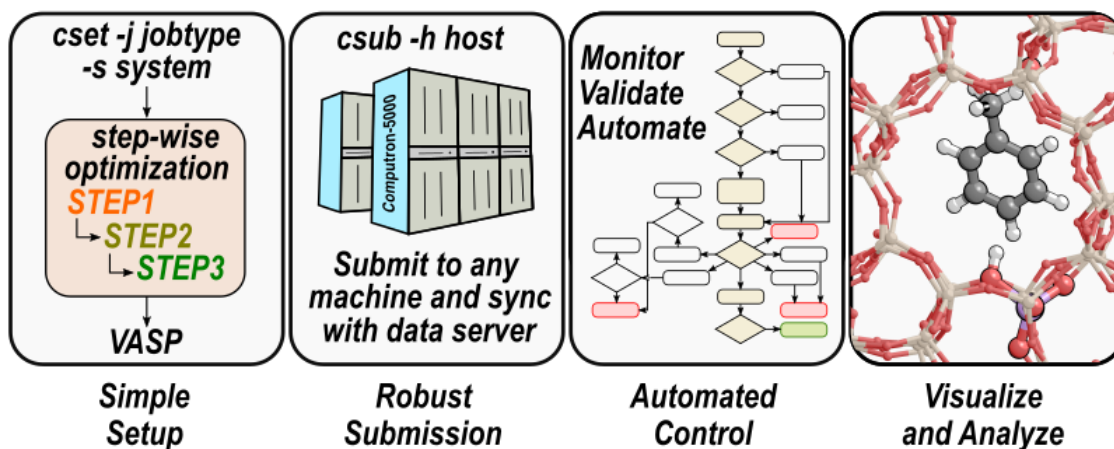
Pavlo Kravchenko,¹ Craig Plaisance,² and David Hibbitts*¹

¹*Department of Chemical Engineering, University of Florida, Gainesville, FL 32611*

²*Department of Chemical Engineering, Louisiana State University, Baton Rouge, LA 70803*

Corresponding author e-mail: hibbitts@che.ufl.edu

Computational Catalysis Interface (CCI)



CCI Improves the Efficiency and Ease of Catalytic Studies

Abstract

Advances in computational resource availability and atomic understanding bring new challenges to studying chemical reactions in sufficient detail—particularly in research of catalytic systems. Here, we present a new interface for density functional theory (DFT) studies—the computational catalysis interface (CCI). CCI simplifies DFT studies and enables multi-step and high-throughput studies with additional automation. Calculations can be set up using descriptive language, allowing users with a wide range of backgrounds to rapidly begin performing DFT calculations. CCI is designed to operate from a centralized data server connected to all accessible compute servers with automated file management between machines. Multi-step calculations with a gradual increase in accuracy increase CPU-efficiency by up to an order of magnitude over single-step calculations. Structure manipulation tools allow users to add or remove adsorbates from catalysts, modify chemical fragments, catalyst composition, and the location of reactions on catalytic surfaces. Specific tools have been developed for modeling metal surfaces and clusters as well as a suite of tools for tackling the configurational complexity of reactions occurring within zeolite micropores. These structure manipulation tools also allow users to generate structures from previously converged calculations by systematically altering the catalyst, reactants, and/or environment (i.e., adding solvent or altering co-adsorbate coverages), which can significantly accelerate rigorous DFT studies. The calculations are automatically monitored and emails are sent to the user when they begin to diverge, fail, or finish. These emails can be customized to contain relevant information and links to a web-based graphical user interface (GUI) that provides a 3-D visualization of structures and movies for convergence trajectories, reaction pathways, and vibrational modes. This web-GUI also allows users to start calculations directly using context-specific commands. CCI also facilitates dissemination of data and interfaces with the POV-Ray utility to generate high-resolution publication- and presentation-quality images. These tools are described here in detail with several examples of reactions and catalysts to show how CCI significantly expedites computational catalysis studies.

1. Introduction

Theoretical understanding of atomic-level interactions has increased over the last three decades with increasingly available computational resources to implement ab initio codes to analyze these interactions. These analyses from first-principles provide rigorous, quantum-level analysis of chemical properties of important and ubiquitous systems. Density functional theory (DFT) is one of the most robust and widely used of these methods for predicting material properties, system energetics, and reaction kinetics. For complex reactions, dozens to thousands of elementary reactions and configurations need to be simulated to effectively understand the reaction network and obtain a desired structure-function relationship. The large numbers of calculations that modern computational resources can achieve require an interface to simplify the setup, submission, and analysis of DFT calculations. To achieve these tasks, the interface must incorporate heuristic approaches to structure manipulation and submission, and to improve on calculation efficiency and analysis.

We present a new open-source interface, the computational catalysis interface (CCI) that is designed to improve the workflow for kinetic analyses by simplifying calculation setup and analysis, improve calculation efficiency, and provide multi-platform visualization. CCI currently interfaces with VASP but is designed around a framework that makes it extendable to other DFT/QM codes or even other input/output-based software such as kinetic Monte Carlo codes. CCI is designed for group level research; group-wide settings and tools are controlled by group “leaders” and individual settings controlled by “users”. Tools and settings for each type of system or project are specified at the group and individual levels and build up to a larger framework designed to rapidly set up, submit and monitor, and analyze numerous VASP calculations. The process for calculation setup has been distilled down to a single command by segmenting routines for file manipulation and settings for VASP optimization into modular components that are combined by CCI. Submission to external machines has been similarly simplified to a single command by specifying host settings at the group level. Calculations are monitored automatically once submitted, prompting the user via email whenever calculations finish or are not running properly. All calculation results for running, failed, or finished jobs can be analyzed using a web-interface that also provides options to run additional jobs directly from the web-page. CCI is written in a combination of BASH, Python, and Fortran, with languages chosen for each command line tool based on requirements (i.e., heavy file handling is suited well for BASH scripts, while mathematical manipulations are reserved for Python scripts or, if necessary for speed, Fortran binaries).

Here, we discuss the methods for improving user workflow for theoretical investigations of catalytic systems. Setting up VASP calculations is simplified by compartmentalizing settings for each system, jobtype, and functional. Calculation efficiency is improved bigly by using multi-step schemes—reducing computational time by a factor of 7 for some systems without sacrificing accuracy. Submission is similarly simplified by providing preset host-specific scripts that submit and monitor calculations without user interaction, while notifying the user of any jobs that converge or require attention via email. Calculations are automatically monitored and pertinent data can be automatically exported to external databases for analysis, reducing user tedium by reducing the need for manual interaction. Analysis is improved through the use of a web-interface that has modules displaying settings, energies, reaction movies, and job control allowing users to submit additional jobs directly from the webpage. CCI also has tools designed to systematically generate or manipulate structures which can, for example, be used to probe effects of coadsorbate coverage on metal systems or reorient species in zeolite pores. CCI removes existing bottlenecks common within computational catalysis and simplifies the process of rigorous examination of complex catalytic systems and reaction networks with the ultimate goal of ending user tedium.

2. Calculation Setup and Efficiency

A VASP optimization requires a minimum of four files: the POSCAR, POTCAR, INCAR, and KPOINTS files. The INCAR contains nearly all VASP directions including settings specifying the algorithms used for the XC functional, and electronic and structural convergence. The POTCAR contains pseudopotentials for each element. Lastly, the KPOINTS file contains the K-point mesh for the sampling method of the Brillouin zone. The POSCAR is supplied by the user or created by CCI and contains structural information—locations and identities of atoms and dimensions of the unit cell for the calculation. Calculations are set up using the **cset** command in a folder with only a pre-existing POSCAR file; **cset** automatically creates the INCAR, KPOINTS, and POTCAR using simplified options (Fig. 1). The INCAR settings for job type (e.g., optimization, nudged elastic band, dimer) and calculation accuracy are defined by the **-j** option (see Fig. 1 for examples). This option also specifies how many sequential calculations, or steps, the job will take to converge. The functional flag, **-f**, specifies the exchange correlation functional settings in the INCAR. The system flag, **-s**, specifies the INCAR settings that correspond to electronic structure convergence for different types of systems (i.e., gaseous molecules or radicals, metal surfaces or bulk, ceramics, or zeolite/zeotype systems, Fig. 1). The potentials flag, **-p**, specifies the pseudopotentials (e.g., PAW, USPP). Non-standard POTCAR files (“hard” or “soft”) can be specified with the **-e** option. The KPOINTS file specifies the K-point mesh¹ for the VASP calculation. This mesh can be specified with **-K** and different meshes can be specified for multistep calculations. When this is not specified, a K-point mesh is automatically generated based on the known geometry and electronic properties of the system. The system, job type, functional, and potential settings are required in **cset** for every VASP calculation. Many optional settings are available with a single flag in the **cset** command (Table 1); however, any other INCAR setting can be passed through **cset** for any step of the calculation using the **--INCAR** flag. This framework allows any type of VASP calculation, including multi-step calculations, to be set up with any settings by running a single command.

The optimal use of CCI involves using a shared directory to store group-wide settings and defaults. This shared directory, referred here as *cci_home*, includes the interface codebase, INCAR fragments (reassembled later) and other files that are necessary for CCI. The codebase is stored in *cci_home/scripts* with executable files (and compiled binaries in some cases) stored in *cci_home/bin*. Host-specific information, including scheduling script headers (e.g., SLURM specifications) and host-specific compiled binaries (e.g., the VASP code itself) are stored in *cci_home/hosts*. POTCAR files are stored for each family of exchange-correlation (XC) functional in *cci_home/potentials*, (n.b., VASP licensing standards mandate that we cannot provide VASP source, compiled binaries, or POTCAR files). Finally, *cci_home/runfiles* contains the information needed to set up the INCAR and KPOINTS files for a VASP simulation. Critically, all settings and defaults in *cci_home* can be overwritten by user-specific files (referred to here as *cci_user_home*) that are organized in the same structure as the shared resource. User-specific CCI does not need to be complete—i.e. individual scripts, INCAR settings, etc. can be specified by a user and all other settings will default to the shared resource specifications. The settings for XC functionals available in VASP are stored in *cci_home/runfiles/functional*. The pieces contained in these settings files are used to build the required files for a VASP calculation. All settings for each system are stored in *cci_home/runfiles/system*.

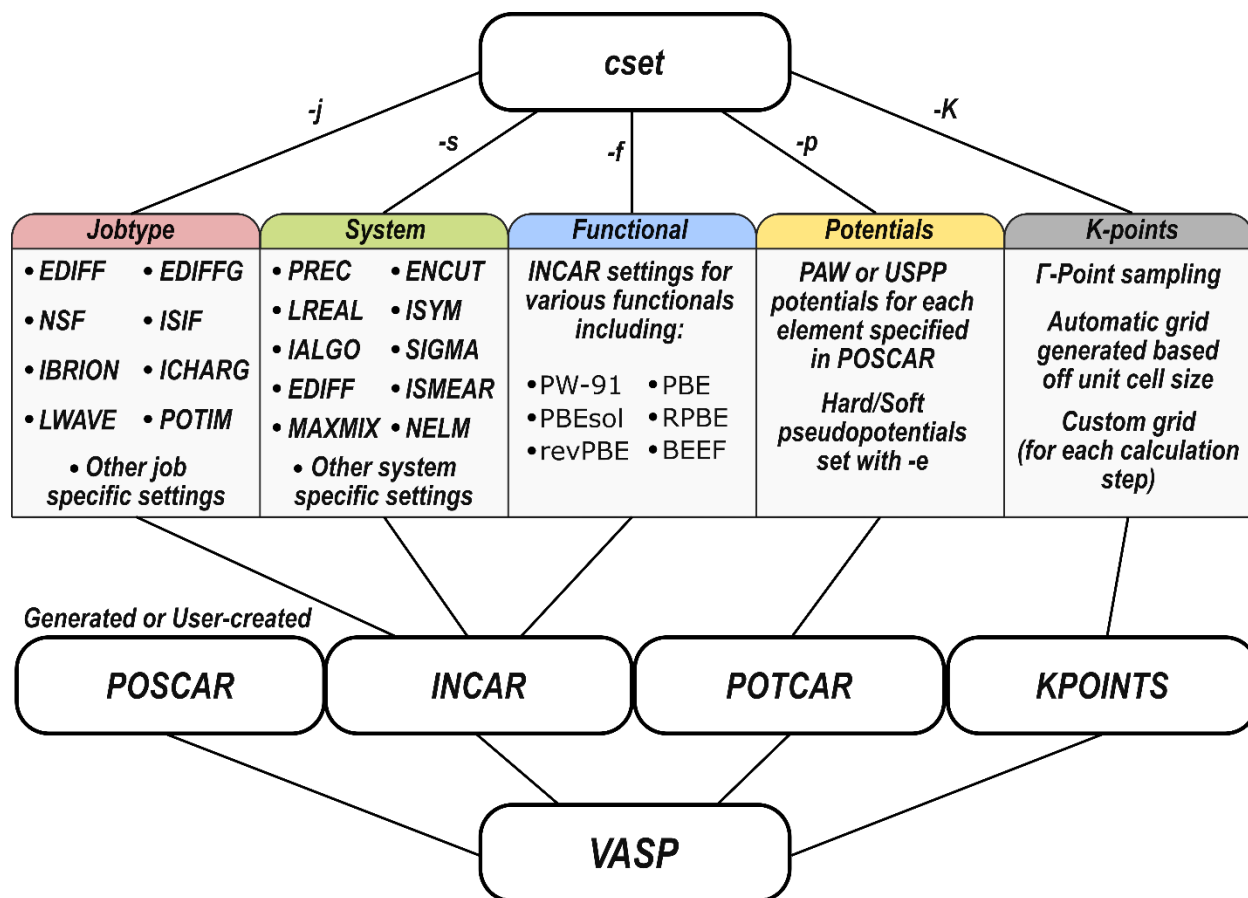


Figure 1. Overview of the how **cset** creates the input files for a VASP calculation, and a sampling of what VASP settings are changed.

Table 1. Common optional settings included in **cset**

Short option	Long option	Description
-m	--spin	Runs spin-polarized calculation
-M	--MAGMOM	Specifies magnetic moments
-U	--Hubbard	Enables DFT+U ² and sets the U parameter
-D	--Dx	Sets dispersion corrections using DFT-D2, ³ -D3, ⁴ or -D3BJ ⁵
	--INCAR	Overwrites or adds any INCAR setting. For example: --INCAR EDIFF=1E-7 (overwrites default)
-B	--Bader	Runs Bader charge ⁶ analysis when calculation finishes
-Q	--Quambo	Runs QUAMBO ⁷ analysis when calculation finishes

Users can create all input files necessary for a VASP calculation by using **cset**—a single, intuitive command—without having to manually copy, paste, edit previously used input files with. Repeated use of **cset** with each of its required flags within a project (where most optimizations use the same settings) can be tedious; default **cset** arguments can be specified in CCI with a file storing specifications “project-wide” (distinct from system-wide or user-wide settings). These *proj_specs* files typically store **cset** arguments for optimizations; NEB and dimer transition state search algorithms; and frequency calculations. These files are stored at any point within your directory hierarchy, and multiple files can be present within your path. Starting from your home directory, CCI

looks for your **cset** arguments (and other project settings) and files closer to your calculation directory will take priority over higher-level files in the case of duplicate settings. This means that individual calculations can be setup by simply running **cset** without specifying any settings. Many commands read the project specs file when setting up and submitting calculations, as discussed later.

Optimizations, including those related to reaction paths (NEB)^{8,9} or transition states (Dimer)¹⁰, are carried out within DFT methods until a desired energy difference or atomic force has been reached in an iterative scheme. Force convergence of atomic structures is more common within computational catalysis and forces ≤ 0.05 eV/Å are generally sufficient for reliable energetics. Accurate computation of these forces, however, require accurate wavefunction convergence criteria (energy differences within self-consistent field (SCF) cycles $\leq 10^{-6}$). Furthermore, for systems with long-range electron entanglement, such as metallic systems, energies at finer K-point meshes are desired, and calculation times scale linearly with the size of the K-point mesh. A “single-step” calculation to achieve these goals is not computationally efficient compared to converging a calculation in multiple steps that progressively increase the accuracy and computational expense of DFT settings as the structure converges towards its final state. The optimization of CH₃-NH₂ on a 4 layer 3x3 Ir(111) metal surface illustrates this: a single-step optimization requires ~12 hours (Table 2), while a three-step approach (Table 2) converges the structure in ~1.6 hours, or 7.5× faster than the 1-step method. This approach starts with a truncated fast-Fourier transform (FFT) grid (PREC flag in VASP) and uses a modest convergence criterion for the SCF cycle (EDIFF of 10^{-4} eV instead of 10^{-6} eV). Then in a second step, the structure is further optimized with an untruncated FFT grid (PREC=.ACCURATE.) and a high SCF cycle convergence criteria (10^{-6} eV) calculation. Finally, a third step consists of a single-point energy calculation at a finer K-point mesh. This three-step scheme results in an energy < 0.01 eV different than the single-step calculation, an error well below kT, and is > 7× faster for this system. This multistep approach accelerates DFT calculations for all systems we have examined, and benefits systems with non-Γ sampling of the Brillouin zone the most. Zeolites, for example, are modeled with only Γ-point sampling and calculations in zeolites are not accelerated by more than 3× by using this method.

Table 2. Multistep optimization schemes for a high accuracy calculation of CH₃NH₂ on a Ir metal surface using the RPBE exchange-correlation functional.

Optimization Scheme	Step	PREC	EDIFF	EDIFFG	KPOINTS	Step time (h)	Total time (h)	Energy (eV)
high_1step	1	ACCURATE	10^{-6}	-0.05	8×8×1	12.02	12.02	-316.2875
high_2step	1	ACCURATE	10^{-4}	-0.05	4×4×1	5.26		-316.4193
	2	NORMAL	10^{-6}	--	8×8×1	0.28	5.54	-316.2911
high_3step	1	NORMAL	10^{-4}	-0.10	4×4×1	0.56		-316.3962
	2	ACCURATE	10^{-6}	-0.05	4×4×1	1.90		-316.4204
	3	NORMAL	10^{-4}	--	8×8×1	0.25	2.71	-316.2914
high_3step_alt	1	NORMAL	10^{-4}	-0.05	4×4×1	1.07		-316.4212
	2	ACCURATE	10^{-6}	-0.05	4×4×1	0.23		-316.4204
	3	NORMAL	10^{-4}	--	8×8×1	0.29	1.59	-316.2922

These multistep optimizations, however, double or triple the number of required VASP calculations, but CCI automates these multi-step calculations. When a step finishes, an optional email notification can be sent to users with the results of that step; otherwise, the calculation continues until all steps are complete. The results of all steps are retained for later analysis.

3. Submitting, Monitoring, and Finalizing Jobs

Once set up, VASP calculations are run on supercomputing clusters and synchronized with a data server. The **csub** command in CCI is used to submit calculations to these supercomputers. This command specifies the supercomputer name (host) and allocated resources for a calculation, syncs the CCI codebase between the data server and compute server, verifies that all necessary files are present, validates the structure provided, transfers all necessary calculation files to the computer cluster, and submits the calculation to the relevant scheduling software on that remote machine (e.g., SLURM). Host-specific settings (including parallelization parameters in the INCAR) and files (such as VASP binaries) are stored in *cci_home/hosts*. Scripts to submit, hold, release, and delete calculations are also host-specific as are “header” region of batch scripts that describe scheduling properties (e.g., cpus, nodes, queue, wall-time, scheduling account email and email settings)—an example of such a header region is given in the Supporting Information (SI, Section S1). The **csub** command uses options **-n**, **-p**, and **-t**, to specify the number of nodes, processors per node, and wall-time, respectively. Defaults for each of these options can be set for each host, so only the host (**-h**) needs to be specified. Many additional optional flags override user-specific settings (email, scheduling account) or allow you to batch jobs into sets which are executed in parallel in a single scheduler job (i.e., if a cluster requires using 64-core calculations, you can run eight 8-core calculations in parallel on a single node rather than running less-efficient 64-core calculations). Scheduler jobs (e.g., PBS, SLURM) which initiate, in parallel, the individual VASP calculations, must then be submitted independently with appropriate flags and resource requests. The **csub** command also validates the structure by analyzing the presence of any “bad bonds” (bonds which are < 60% of their predicted distance based on covalent radii combinations). This can be overridden with the **--force** flag. If bad bonds are detected, CCI attempts to fix the bonds by setting reasonable bond lengths based on covalent radii (this can be overridden with the **--no_fix** flag). If the structure cannot be fixed, **csub** aborts submission. A unique job-number (sequence specific to the user) is generated for all calculations. A webpage showing the submitted structure, reaction pathway (NEB), or transition state structure and reaction mode (Dimer) is generated for verification (Section 4). Figure 2 shows a process diagram for the **csub** command. A job is run by a PBS script on the compute server. The PBS script controls a calculation, either individually or as part of a batch. The header of this file contains resource allocations (specified by **csub**) and both host account information and user information on the data server. Once the job or batch is started, the pbs script updates CCI-specific tracking file called *specs* and starts the relevant MPI process (typically VASP) for parallel calculations.

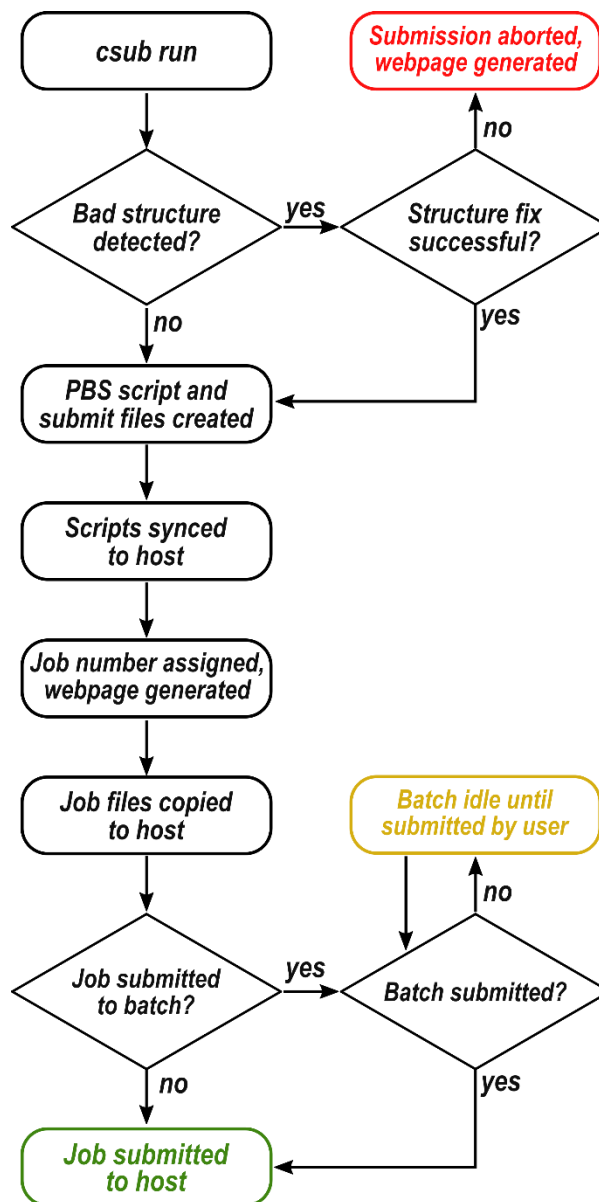


Figure 2. Flowchart outlining job submission processes. Jobs can be submitted individually or batched to run multiple jobs on a single node (optimizing node utilization).

Throughout the calculation, forces are monitored based on two user-defined criteria: a “fail” limit and a “warn” limit. The first may terminate the calculation if the force is extremely high (kill force), indicating a bad input structure or that a bad step within the structural optimization has led to a place unlikely to converge to a satisfactory result in a reasonable amount of computational time. Initial structures often have large forces, so CCI waits until a specific number of structural iterations (e.g., 10–20) have passed before enforcing this limit. Similarly, the “warn” limit will designate a force at which an email will be sent to the user to indicate a potential issue with the calculation. This email will show the convergence progress (up to the problematic iteration) and will include a link to the website results viewer showing structural details. Data from ongoing calculations are periodically copied back to the data server to allow for continuous monitoring while the VASP process is running.

When the calculation approaches the wall-time limit, the pbs script initiates a “soft stop” condition, which stops VASP after the current SCF cycle converges, and (closer to wall-time expiration) a “hard stop”, which stops VASP after the current step within the SCF cycle. Both the soft and hard stops utilize the STOPCAR file. These

protocols aim to stop VASP slightly before the end of the wall-time (at which point the scheduling software will terminate) and allow CCI to prepare the calculation for resubmission and resubmit it if possible. A user only needs to submit a calculation once and it will run until completion or termination without further interaction when using this scheme. When a job is terminated or suspended for whatever reason (e.g., force too high, node failure, exceeding wall-time), the user is notified by email and the latest information is copied back to the data server. Jobs that have been terminated can be easily restarted by the user or can be restarted automatically by CCI for some failure modes. Figure 3 shows a process diagram for automated job monitoring and control processes.

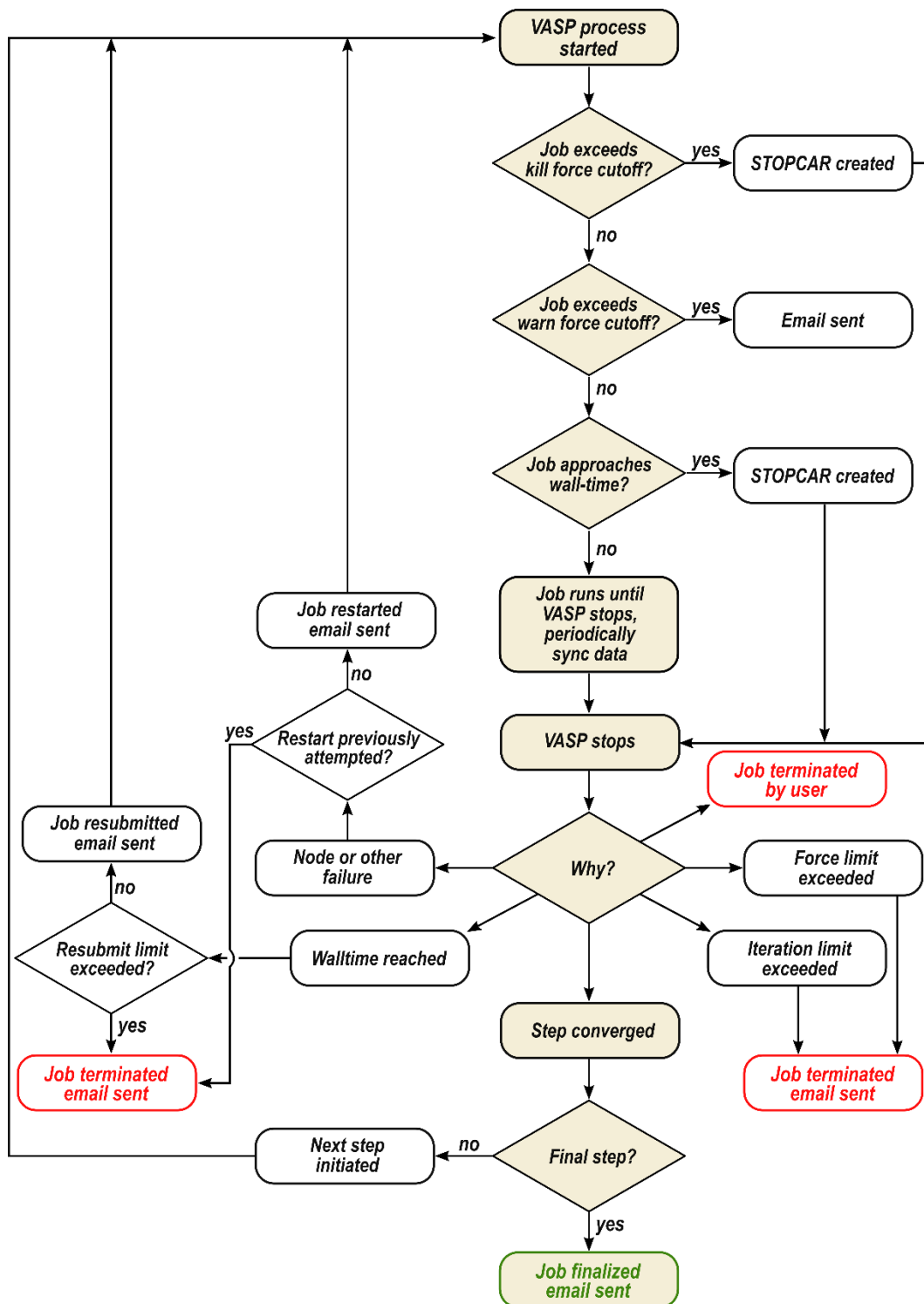


Figure 3. Flowchart outlining automated processes for job monitoring after VASP is executed. The path for a calculation that converges without issue is highlighted. Settings for force limits and wall-time are defined in the `pbs_cci` script. Iteration limit is set with the `NSW` flag in the `INCAR` for each calculation step. While running, jobs are periodically synced with the data server to allow for continuous job monitoring.

Jobs can be monitored in real time and manipulated manually using queue tools configured for each compute server. These queue tools display calculations across all available/utilized compute servers to provide a single interface accessible from the data server. The **q** command displays all idle and running jobs and displays their CCI-specific job number, compute server, a calculation name/label, status, the ID assigned by the scheduler (queue ID), job type, remaining wall-time, and the current force, iteration, and energy of the calculation. The output for the queue can be filtered based on job status (held, idle, running), host, user, or job type. This command **q** can also be used to hold, release, or kill jobs—either individual jobs, all jobs, or any set of jobs filtered by some criteria. This command is also used, within our group, to balance and prioritize calculations across users, projects, and specific calculations. Calculations are initially submitted to compute servers as “held” and then released based on current utilization, PI-designated user/project priorities, and user-specified priorities to allow for effective management of large numbers of queued calculations and the acceleration of critical calculations.

Once a calculation finishes, it is ‘finalized’, which allows for post-VASP tools to be executed in addition to subsequent (dependent) calculations to be submitted. Output files are copied back to the data server and a final email is sent to the user notifying them that the job finished (with relevant information and a link to the web-based results GUI, Section 4). Finalization scripts run additional calculations (e.g., frequency, QUAMBO, or charge analyses), update databases, or run any procedure specified by the user or group leader. A similar and common routine within catalytic studies is to optimize a reactant and product, initialize reaction pathway search (NEB), and refine the identified transition state using the Dimer algorithm (or climbing image NEB)—though this is not the only sequence of serialized calculations used to isolate a transition state. CCI has three specific tools to facilitate these routines: **setup_rxn**, **setup_neb**, and **setup_dimer** (other tools exist for similar serialized calculations such as **setup_FREQ** to initiate a vibrational frequency calculation). The first of these tools (**setup_rxn**) takes a series of structure files from the user and sorts them into *Reactants*, *Products*, and *NEB* folders. The first and last structures are placed into the *Reactants* and *Products* folders, respectively; within these folders, **cset** and **csub** are run without arguments, thus using default settings from the *proj_specs* file to set up and submit these two calculations. Upon completion, **setup_neb** takes these converged structures, combines them with any intermediate structures previously provided, and initiates and submits an NEB calculation based on defaults from **cset** and **csub**. Finally, **setup_dimer** can initiate a Dimer calculation to isolate the transition state from the NEB output; this command can either interpolate between images based on predicted maxima along the reaction coordinate diagram (similar to the *neb2dim* script VTST provides)¹¹ or, with different arguments, begin from a specific image of the NEB. This command can also be used to start calculations from on any iteration of the NEB, which is very useful if the NEB method fails to converge because users can specify the iteration with the lowest maximum force. This sequence of **setup_rxn** → **setup_neb** → **setup_dimer** can be executed “manually”, in a completely automated fashion (via calculation finalization described above), or via the results website (as discussed in Section 4) where structures can be user-validated prior to NEB and Dimer submission.

4. Website Visualization and Image Generation

CCI includes a built-in web interface for viewing structures and completed calculations. Each page is modular based on the calculation type and convergence (i.e., a single webpage source is used for all calculations with modules displayed based on context). Each module can be hidden or shown with a toggle (+/−) (Fig. 4). For each

calculation, information is shown such as the job-number, data server directory (truncated), status, iteration, force, and energies are reported for analysis. The final (or initial) structure can be downloaded from the webpage. The structure is rendered using Jmol¹² and can be interacted with using keyboard and mouse controls or interactable toggles for more precise control.

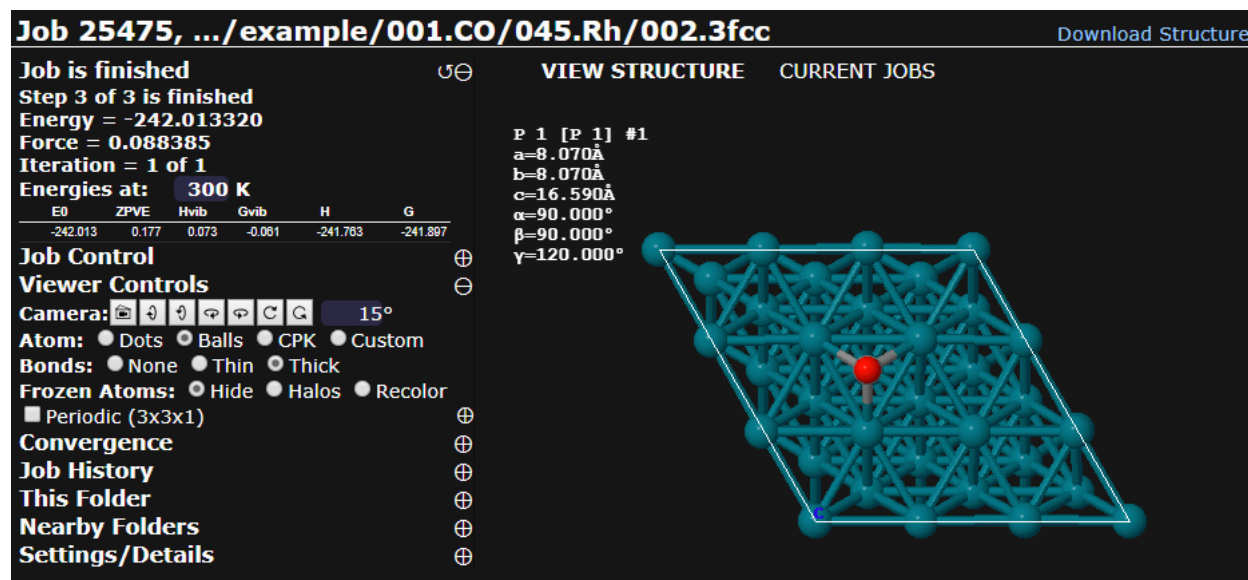


Figure 4. Sample webpage showing the toggleable modules of an optimization calculation. The structure can be viewed in different styles and the converged structure can be downloaded directly from the webpage.

Viewing styles, unit cell periodicity, or unit cell offsets can also be changed using options under the viewer control toggle (Fig. 5a), with more advanced options available in the Jmol console for advanced users. Defaults for each of the viewer settings are customizable and are set at the project level in the *proj_specs* file. The Job Control module (Fig. 5b) allows one to run a variety of CCI commands directly from the web-based GUI. These commands are provided based on context and, for security purposes, only options of the commands are user-editable (i.e., arbitrary commands cannot be entered). A common use of these commands is shown in Fig. 5b, in which the user can set up dimer calculations or optimizations from a NEB pathway, or they can “shorten” an NEB between two local minima along the pathway (this will initiate a new reactant, product, and ultimately NEB calculation). Web-users can directly kill calculations that are in progress, perhaps if they were warned via email of poor convergence. Webpages showing the initial structure, the results of intermediate steps (in multi-step calculations) or previous runs (prior to resubmission) can be viewed in the Job History module. Links to results websites for related calculations (such as those that share the same parent folder in the directory tree) links to nearby jobs in the same calculation folder or nearby parent folders are also available under additional toggles (Fig. 5d,e). Lastly, the Settings/Details module displays the full directory, the arguments used for **cset** and **csub** to set up and submit the calculation, and information regarding the POSCAR, INCAR, KPOINTS, and POTCAR VASP input files. The latter three are all displayed in a manner to demonstrate how parameters (such as INCAR settings) are set for each step in a multi-step calculation.

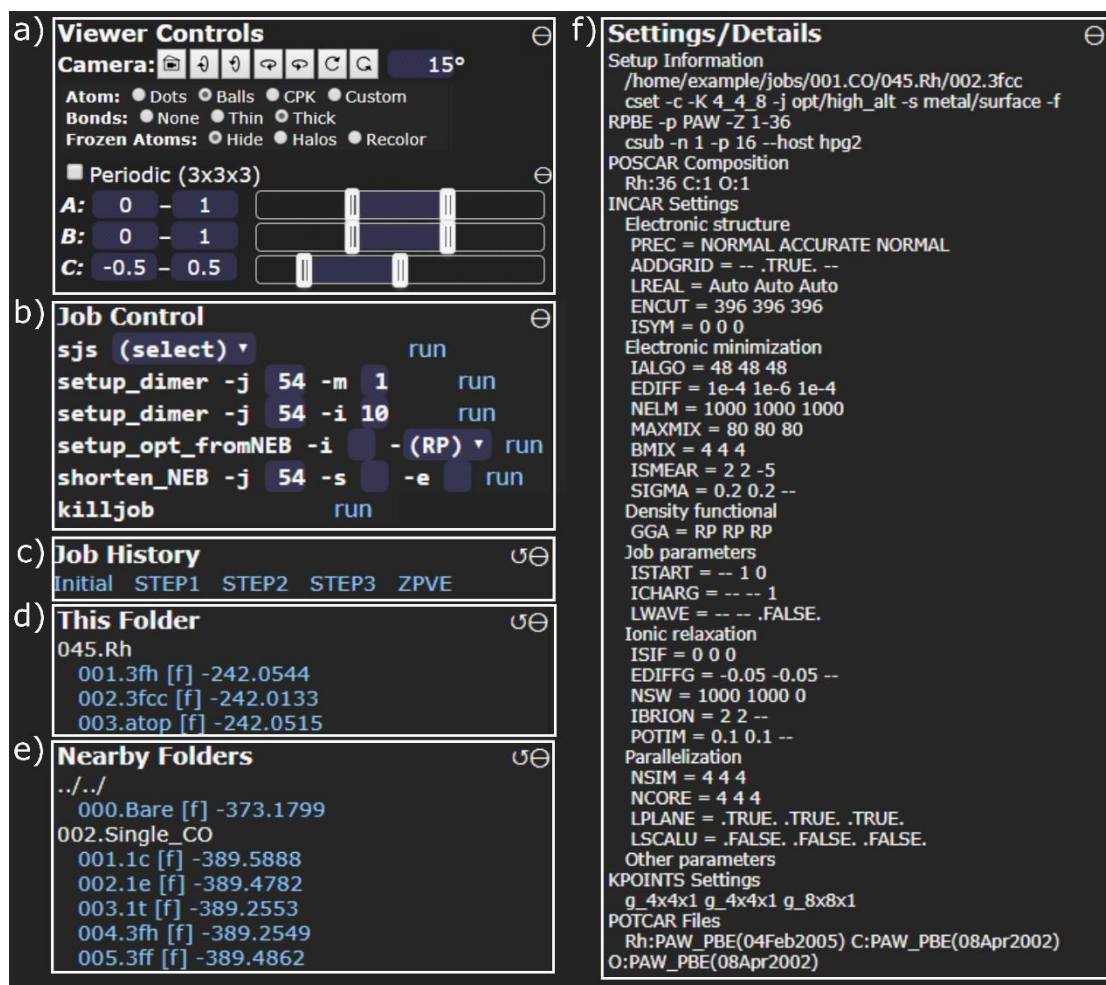


Figure 5. Modules available for every job type. All jobs have modules for: a) viewer controls to manipulate the camera, view styles, visibility of frozen atoms, and periodic views to view fractional or supercells (with any offset); b) job control to change the status of a job post-run, run complementary jobs such as dimers (shown) or frequency calculations (not shown), or to kill an active job; c) job history showing intermediate convergence steps and additional calculations (like frequency or charge calculations); d) nearby jobs run in the same subfolder or in e) nearby parent folders; and f) a table of information about the calculation including all input file settings.

Each job has unique modules, in addition to the ‘global’ modules. For optimization calculations, a convergence trajectory (movie in Jmol) can be viewed for any VASP calculations which had multiple structural iterations (Fig. 6a). NEB calculation results include a reaction coordinate diagram showing the energy of each image and a movie showing the minimum energy pathway for the reaction (Fig. 6c). Dimer calculation webpages show an energy diagram with the reactant, product, and transition state energies for the reaction, as well as a movie showing the reaction mode or the reactant, transition state, and product structures (Fig. 6b). When Jmol movies are available, controls appear to allow users to pause, play (in either direction), step forward/backward, or go to the end/beginning of the structural movie. Additional options to visualize charge are also available for each structure or job that had a charge analysis performed (Fig. 6c). Frequency calculation result pages show the frequencies for each calculated mode and clicking on such frequencies initiates movies showing the vibrational mode (Fig. 6d). Temperature-corrected enthalpies and free energies (at 1 bar) are also displayed for finished vibrational frequency calculations or optimizations for which a vibrational frequency calculation is present. Low-frequency ($\leq 100 \text{ cm}^{-1}$) modes have documented inaccuracies and are often inappropriate for harmonic oscillator

entropy estimates; thus, frequency cutoffs can be specified, beneath which a representative value (e.g., 60 cm^{-1}) can be used instead as done previously for zeolite catalyzed dehydration and condensation reactions.^{13,14}

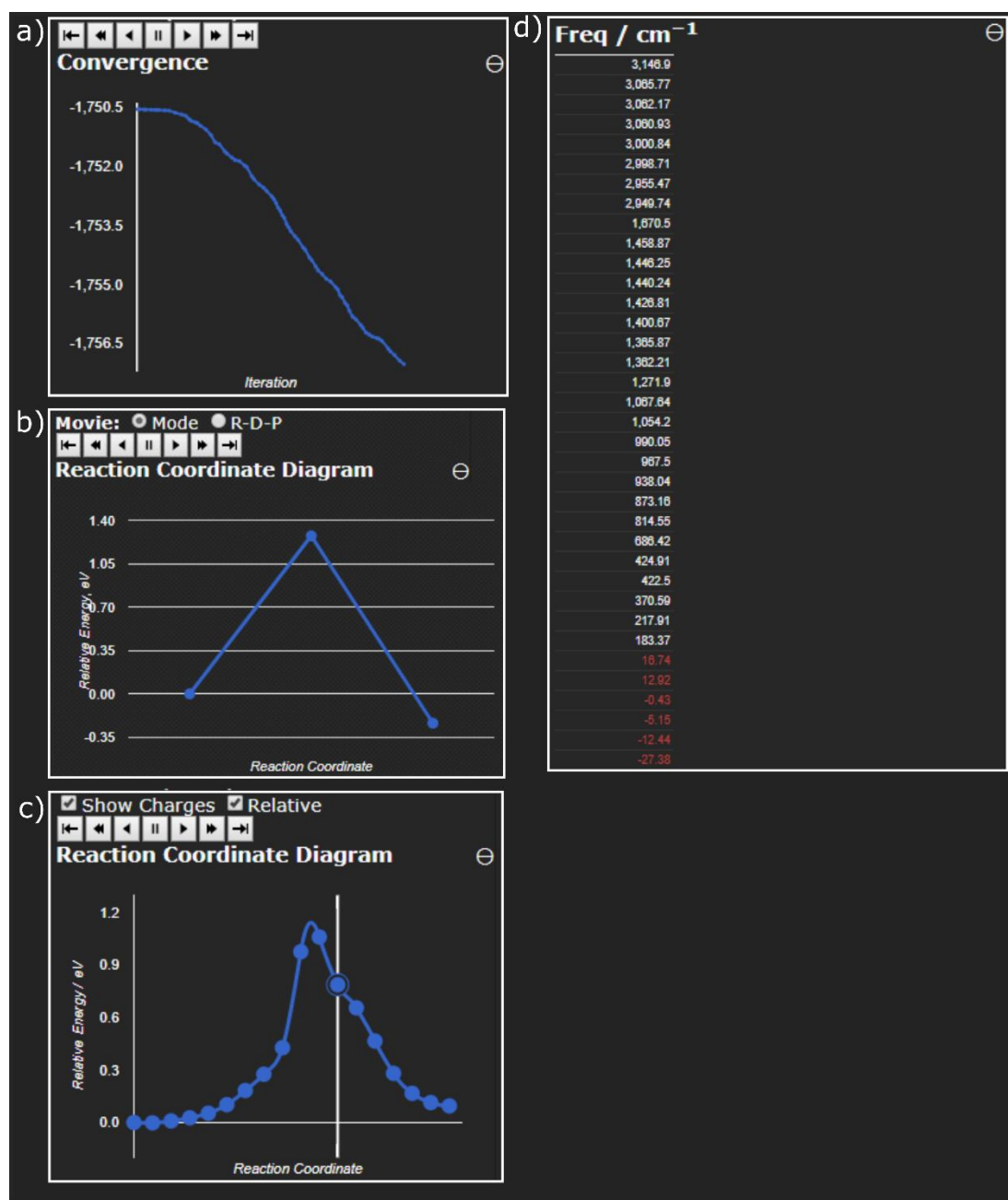


Figure 6. Job specific modules for a) optimization convergence; b) dimer calculations showing reactant, transition state, and product energies; c) NEB reaction coordinate diagrams; and frequency tables. Each point corresponds to a structure and can be viewed as a movie on the webpage. Each individual vibrational frequency can also be viewed as a movie.

Visualizing structures and reactions can be conveniently done through the webpage, but CCI also has tools to create high resolution stylized images for visualizing. These high-quality image tools utilize the POV-Ray package¹⁵ to render a scene with raytracing. The atom styles can be modified so that different styles appear for atoms that are part of the system (e.g., metals or zeolites) or are adsorbates. The atoms are assigned as systems or adsorbates automatically by CCI, and additional styles can be set for user-labeled atoms. Viewing windows can

be set by specifying the virtual camera's focal point, tilt, and zoom for any specified number of unit cells (or fractional cells). Periodic atoms can be shown or hidden based on type (for example, hiding periodic adsorbate atoms to give the impression of an extended metal surface). All settings for these images are stored in header files which provide default settings for every type of system. For NEBs, dimers, or individual vibrational modes, multiple images can be created and joined together to create high framerate movies at the same level of quality and resolution as the images. Examples of POV-Ray images are shown below in Section 5 and within the SI, Section S2, as well as movies uploaded as supporting documents.

5. Structure Generation and Manipulation

CCI has a host of structure manipulation tools that enable rapid structure generation while minimizing the need for “hand-made” structures by the user—thereby minimizing user error, accelerating the setup of DFT calculations, and providing more accurate initial guesses for structures. Catalyst-specific tools are available for nearly all catalytic systems but focusing on metal surfaces/particles and zeolite/zeotype materials with some tools developed for oxides and ceramics. Many other tools focus on the manipulation of organic moieties and thus are applicable to any catalytic reaction. These tools often use several core commands:

- **connectivity** determines which atoms are bound to one another and identifies bonds that are too short (indicative of a potentially problematic structure).
- **bond_alter** sets or measures the distance between 2 atoms, the angle between 3 atoms, or the torsional angle between 4 atoms. Points or vectors in Cartesian or direct coordinate spaces can be specified instead of atom indices.
- **atom_add, atom_del, atom_elem** adds, deletes, or alters the identify of an atom.
- **cleave_surface** can be used to generate a surface slab model for any Miller index (*hkl* or *hkil*) of specified unit cell size, slab thickness, and vacuum thickness for any crystallographic information (.cif) file

These generalized commands—among many others—are used for a suite of structure manipulation tools and, when combined, accelerate work performed prior to calculations submission.

Metal surfaces can be generated for (111) and (100) surfaces of FCC metals (e.g., Pt), (0001) surfaces of HCP metals (e.g., Ru), and (110) surfaces of BCC metals (e.g., Fe) using **build_metal**. The unit cell size, vacuum thickness, slab thickness, and the number of metal layers fixed in bulk crystallographic positions can all be specified. Adsorbates can be added to any site (e.g., atop, bridge, 3-fold) or any combination of sites (see Fig. 7 for examples) to close-packed surfaces with **add_adsorbate** (Table 3 shows adsorbates which can currently be added using this command). These have been used to examine the binding energies of CO* on Ru(0001) surfaces,¹⁶ H* on Pt and Ir(111),^{17,18} and NO* on Pt(111).¹⁷ Cubo-octahedral FCC metal particles can be generated with **build_particle** with user-specified shape and size (users specify the size of (111) and (100) crystal facets and thus can build a wide variety of particle models - see Section S3 in the SI for examples). Tools also exist to add adsorbates to metal particles based on binding site ensembles (i.e., adding a H* atop to all corner atoms on a 538-atom cluster). These tools were used to examine how H₂ chemisorbs (as H*) on Pt and Ir nanoparticles by calculating ensembles of H₂ chemisorption at sub- to supramonolayer coverages.¹⁹ Thousands of structures can be programmatically generated without any manual setup of structures, greatly reducing user time and error.

Table 3. All adsorbates that can be automatically added to sites on a metal surface or particle.

-C	-CO	-N	-O
-CH	-H	-NH	-O-H ₂ O
-CH ₂	-H ₂ O	-NH ₂	-O-OH
-CH ₂ O	-H ₃ O	-NH ₃	-OH-H ₂ O
-CH ₃	-HO-H-OH	-NH ₃ -OH	-S
-CH ₄	-HOH-OH	-NO	-F

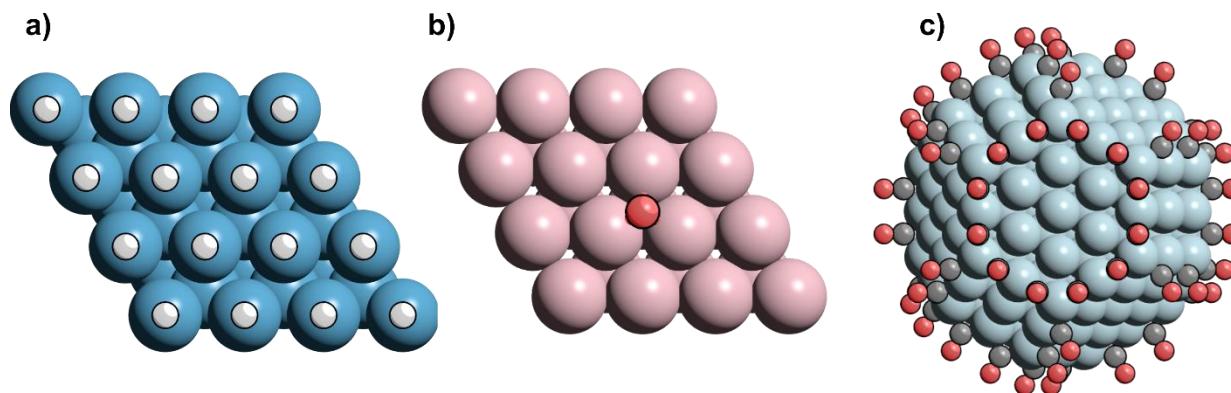


Figure 7. Example structures generated by **build_metal**, **build_particle**, and **add_adsorbate** tools. a) A hydrogen covered Pt(111) surface, b) an O-atom adsorbed on a Co(0001) surface, and c) a Rh₂₀₁ particle with CO covering every bridging edge site.

Surface reactions, once converged for one catalyst model, can then be modified by adding co-adsorbates, as just described, to examine coverage effects. This has been used to examine coverage effects for C–C hydrogenolysis on Ir surfaces¹⁸ and NO reduction on Pt surfaces¹⁷ through a combination of **add_adsorbate** and **atom_del** to fill and remove spectating H* (and NO*) adsorbates, respectively, in a range of coverages and adsorbate arrangements. Alternatively, the metal catalyst can be changed instead of the adsorbate structure (using **metal_convert** tools, the first structure modification tool written for this package)—a scheme that has been used successfully to examine multiple metals of the periodic table for alcohol oxidation,^{20–22} alkane hydrogenolysis,²³ and O* and OH* reactivity.²¹ Similarly, converged structures of adsorbates, reaction pathways, and transition states can be relocated from one catalyst model to any other, such as transferring surface model results to a nanoparticle model (**surf_to_particle**, Fig. 8). These tools have been successfully used to examine alkane hydrogenolysis¹⁸ and CO activation during Fischer-Tropsch synthesis.^{24–27}

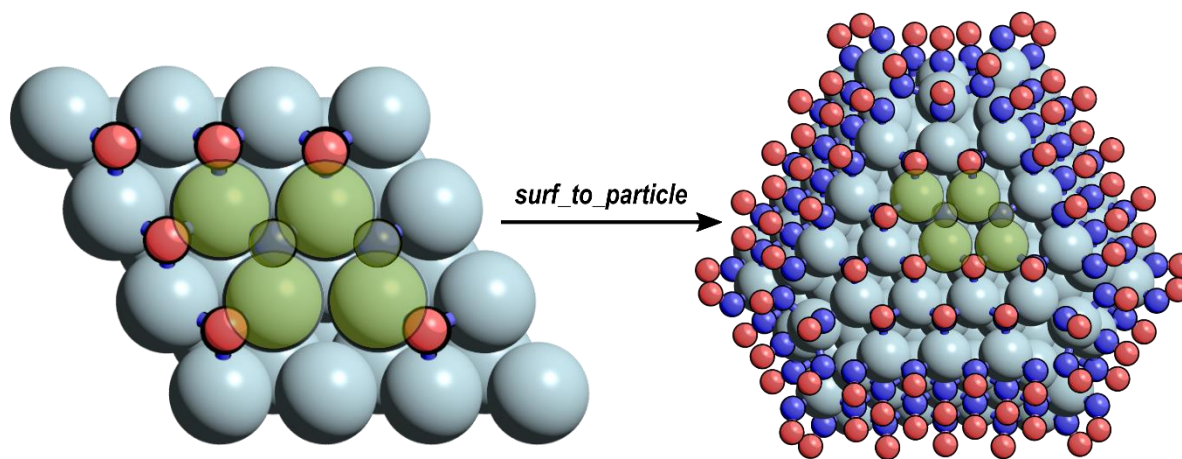


Figure 8. Atoms that are relocated from a surface model to a NO*-covered nanoparticle model using **surf_to_particle** tools.

Structures of adsorbates or molecules, regardless of the catalyst system, can be functionalized or defunctionalized through the **H_to_frag** and **frag_to_H** commands (see Table 4 for all substitutional groups). These tools can be used to take converged structures (adsorbates, reaction pathways, or transition states) and alter the reacting species. For example, starting with the methylation of pentamethylbenzene to form hexamethylbenzene, the structures can be de-methylated to examine all ~20 methylation reactions (such as benzene to toluene), greatly accelerating the set up and convergence of the CCI-modified structures.²⁸ These tools have also been successfully used to study the effect of chain length on alkane hydrogenolysis,^{29–31} and during Fischer-Tropsch synthesis,²⁴ and for C–H and O–H bond activation on metal surfaces.²²

Table 4. All possible fragments in CCI. Any fragment can be converted to any other fragment via –H.

–CH ₃	–CHO	–CHNH	–H	–NH ₂
–CHCH ₂	–COOH	–CN	–OH	–NNH
–CCH	–CHOCH ₂	–NCO	–OOH	–SH
–C ₆ H ₅				

For many investigations of catalyzed reactions, some atoms in the catalyst model are constrained. For surface and half-particle models systems, some number of the bottom-most layers are typically constrained in bulk crystallographic positions (done in CCI with the **fz_layers** command, which can freeze atoms below a certain fractional coordinate or based on atomic layers). Individual atoms can be constrained using the **fz_atoms** command (which can also unconstrain atoms). Inputs for these constraint commands can also be stored in *proj_specs* files which will then be applied to all relevant structures within that project or sub-project. For example, we constrain the bottom two layers of 4-layer metal slabs for optimizations and we constrain all metal atoms for frequency calculations metal surface calculations. No atoms are constrained during optimizations of zeolite-catalyzed reactions, while all atoms are constrained except for the adsorbate and the acid site during a frequency calculation.

For zeolites or zeotype materials, tools have recently been developed to aid in examining the 3-D micropores and diverse topology of acid sites. Reactions in proton-form zeolites occur near Brønsted acid sites and zeolite

frameworks have multiple unique locations for acid sites or have multiple acid sites in close proximity. Modeling a reaction at every possible acid site or combinations of sites in CCI is aided by **zeo_relocate** tools, which take converged adsorbate structures at one acid site and relocate them to any other site within the framework (or to a new framework). Similar tools have been created to add “pre-converged” adsorbate structures in any arrangement around acid sites while rejecting structures that collide with the framework. These tools were successfully used to characterize acid strength in the CHA framework by adsorbing NH_4 in various configurations around multiple acid sites.³² Adsorbates can also be relocated between O-atoms on the same acid site using **zeo_relocate**.

The orientation of reactant, product, and transition state structures within a zeolite pore can have significant effects on the energy of the state^{28,32}—suggesting that single DFT calculations may not provide reliable energies. This can be overcome using techniques such as ab initio molecular dynamics (AIMD),^{33–35} however, AIMD calculations are computationally expensive and it is not feasible to investigate large reaction networks with these techniques. One method to overcome this limitation, is to systematically reorient and re-optimize adsorbates and transition states in zeolites by rotating species based upon their interaction with the Brønsted acid site. Species that strongly interact with the Brønsted acid site via hydrogen or covalent bonding (e.g., CH_3OCH_3) are rotated by altering the orientation with respect to the Brønsted acid site (Fig. 9a). Adsorbates that interact nonspecifically with the Brønsted acid site are rotated with about the a-, b-, and c-axes of the unit cell (Fig. 9c). Transition state complexes containing multiple fragments that can rotate about breaking bonds are rotated about internal coordinates of the structure (Fig. 9b) in addition to other eligible reorientation schemes (such as about the Brønsted acid site or about the a-, b-, c-axes of the unit cell). This reorientation mechanism does not guarantee location of the global minimum; however, it provides insight to adsorbate-framework interactions and generates states lower in energy than initial user-generated structures. **Systematic-reorientation** tools have been developed within CCI to rapidly setup these distinct reorientation structures from an optimized user-generated structure. Structures set up by **systematic-reorientation** tools are rotated based upon their interaction with the Brønsted acid site. Each reoriented structure is automatically checked for ‘bad’ bonding (i.e., states that collide with the framework) and those states are discarded (Fig. SX). The remaining states are automatically submitted for re-optimization, resulting in a streamlined submission and analysis process of ~12–90 DFT-calculated structures for each adsorbate and transition state structure within zeolites.

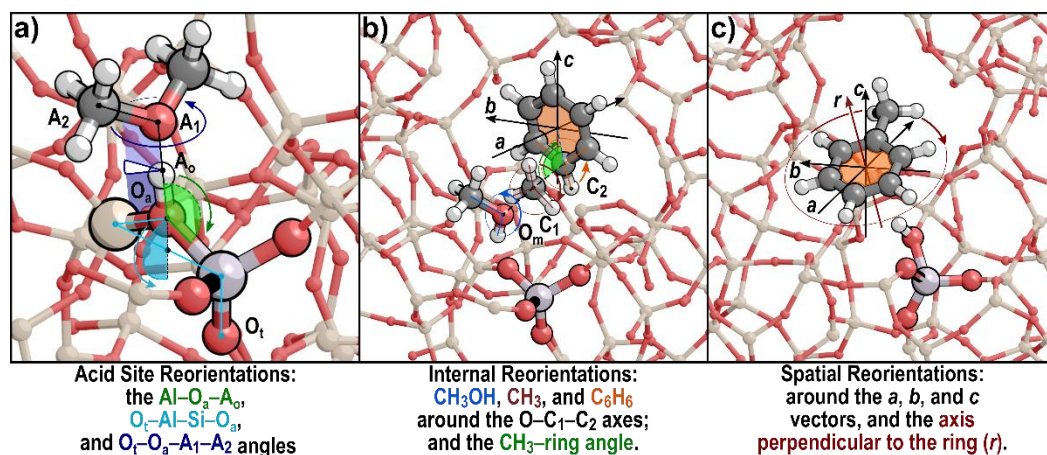


Figure 9. a) Three different reorientations of CH_3OCH_3 about the Brønsted acid site, b) internal reorientations of the benzene methylation transition state where benzene and CH_3OH can be rotated about the breaking bonds and the entire transition state complex can be rotated about the a-, b-, and c-axes of the unit cell, and c) spatial reorientations of toluene about the a-, b-, and c-axes of the unit cell.

6. Conclusions

Here we presented an overview of our computational catalysis interface (CCI) and have described how we have used both generic and specific tools to dramatically accelerate our research over the past ten years. At its core CCI is a framework that categorizes a many settings for systems and calculation types that can be pieced back together to efficiently setup and run calculations. This interface is open-source and fully customizable. It is designed to make computational catalysis more user-friendly and more efficient through heuristic approaches while minimize user error and enabling tools to implement high-throughput structure manipulation to examine and rapidly assess the effects of catalyst composition and morphology as well as reactant substituent effects without using Brønsted Evans Polyani or scaling relationships. Finally, the web-based GUI and POV-Ray tools facilitate visualization of computational catalysis results for validation, instruction, and dissemination. Currently, additional tools for detailed analysis of large reaction networks using kinetic Monte Carlo simulations or prescreening of structures to reduce the required number of configurations with neural networks are being developed. These tools will grant CCI chemical intuition to aid in screening potential improved catalysts.

Acknowledgements

David Hibbitts and Craig Plaisance acknowledge Matthew Neurock for allowing us the intellectual freedom to start and implement this interface at Univ. of Virginia. David Hibbitts acknowledges Enrique Iglesia, Michele Sarazen, William Knaeble, Andrew Jones, and Gina Noh for useful discussion and feedback during their use of CCI tools at Univ. of California–Berkeley. Pavlo Kravchenko and David Hibbitts acknowledge Alex Hoffman, Abdul Almithn, Mykela DeLuca, and Steven Nystrom for their contributions to the Supporting Information, for their support in documenting CCI features, and for their feedback during their use of CCI at Univ. of Florida.

References

- (1) Monkhorst, H. J.; Pack, J. D. Special Points for Brillouin-zone Integrations. *Phys. Rev. B* **1976**, *13*, 5188–5192.
- (2) Himmetoglu, B.; Floris, A.; de Gironcoli, S.; Cococcioni, M. Hubbard-corrected DFT Energy Functionals: The LDA+U Description of Correlated Systems. *Int. J. Quantum Chem.* **2014**, *114*, 14–49.
- (3) Grimme, S. Semiempirical GGA-type Density Functional Constructed with a Long-range Dispersion Correction. *J. Comput. Chem.* **2006**, *27*, 1787–1799.
- (4) Grimme, S.; Antony, J.; Ehrlich, S.; Krieg, H. A Consistent and Accurate Ab Initio Parametrization of Density Functional Dispersion Correction (DFT-D) for the 94 Elements H–Pu. *J. Chem. Phys.* **2010**, *132*, 154104.
- (5) Grimme, S.; Ehrlich, S.; Goerigk, L. Effect of the Damping Function in Dispersion Corrected Density Functional Theory. *J. Comput. Chem.* **2011**, *32*, 1456–1465.
- (6) Bader, R. F. W. *Atoms in Molecules: a Quantum Theory*; (New York: Oxford University Press), Ed.; 1990.
- (7) Lu, W. C.; Wang, C. Z.; Chan, T. L.; Ruedenberg, K.; Ho, K. M. Representation of Electronic Structures in Crystals in Terms of Highly Localized Quasiatomic Minimal Basis Orbitals. *Phys. Rev. B* **2004**, *70*.
- (8) Henkelman, G.; Jónsson, H. Improved Tangent Estimate in the Nudged Elastic Band Method for Finding Minimum Energy Paths and Saddle Points. *J. Chem. Phys.* **2000**, *113*, 9978–9985.

- (9) Jónsson, H.; Mills, G.; Jacobsen, K. W. Nudged Elastic Band Method for Finding Minimum Energy Paths of Transitions. In *Classical and Quantum Dynamics in Condensed Phase Simulations*; Berne, B. J.; Ciccotti, G.; Coker, D. F., Eds.; World Scientific, 1998; pp. 385–404.
- (10) Henkelman, G.; Jónsson, H. A Dimer Method for Finding Saddle Points on High Dimensional Potential Surfaces Using Only First Derivatives. *J. Chem. Phys.* **1999**, *111*, 7010–7022.
- (11) VTST Tools Webpage <http://theory.cm.utexas.edu/vtsttools/> (accessed May 7, 2019).
- (12) Hanson, R. M. Jmol – a Paradigm Shift in Crystallographic Visualization. *J Appl Crystallogr* **2010**, *43*, 1250–1260.
- (13) Herrmann, S.; Iglesia, E. Elementary Steps in Acetone Condensation Reactions Catalyzed by Aluminosilicates with Diverse Void Structures. *J. Catal.* **2017**, *346*, 134–153.
- (14) Zhi, Y.; Shi, H.; Mu, L.; Liu, Y.; Mei, D.; Camaioni, D. M.; Lercher, J. A. Dehydration Pathways of 1-Propanol on HZSM-5 in the Presence and Absence of Water. *J. Am. Chem. Soc.* **2015**, *137*, 15781–15794.
- (15) Plachetka, T. POV Ray: Persistence of Vision Parallel Raytracer. *Proc. of Spring Conf. on Computer Graphics, Budmerice, Slovakia* 123.
- (16) Loveless, B. T.; Buda, C.; Neurock, M.; Iglesia, E. CO Chemisorption and Dissociation at High Coverages During CO Hydrogenation on Ru Catalysts. *J. Am. Chem. Soc.* **2013**, *135*, 6107–6121.
- (17) Hibbitts, D. D.; Jiménez, R.; Yoshimura, M.; Weiss, B.; Iglesia, E. Catalytic NO Activation and NO–H₂ Reaction Pathways. *J. Catal.* **2014**, *319*, 95–109.
- (18) Almithn, A.; Hibbitts, D. Effects of Catalyst Model and High Adsorbate Coverages in Ab Initio Studies of Alkane Hydrogenolysis. *ACS Catal.* **2018**, *8*, 6375–6387.
- (19) Almithn, A. S.; Hibbitts, D. D. Supra-monolayer Coverages on Small Metal Clusters and Their Effects on H₂ Chemisorption Particle Size Estimates. *AIChE J.* **2018**, *64*, 3109–3120.
- (20) Zope, B. N.; Hibbitts, D. D.; Neurock, M.; Davis, R. J. Reactivity of the Gold/water Interface During Selective Oxidation Catalysis. *Science* **2010**, *330*, 74–78.
- (21) Hibbitts, D. D.; Neurock, M. Influence of Oxygen and pH on the Selective Oxidation of Ethanol on Pd Catalysts. *J. Catal.* **2013**, *299*, 261–271.
- (22) Hibbitts, D.; Neurock, M. Promotional Effects of Chemisorbed Oxygen and Hydroxide in the Activation of C–H and O–H Bonds over Transition Metal Surfaces. *Surf Sci* **2016**, *650*, 210–220.
- (23) Almithn, A.; Hibbitts, D. Comparing Rate and Mechanism of Ethane Hydrogenolysis on Transition-Metal Catalysts. *J. Phys. Chem. C* **2019**, *123*, 5421–5432.
- (24) Hibbitts, D. D.; Dybeck, E.; Lawlor, T.; Neurock, M.; Iglesia, E. Preferential Activation of CO Near Hydrocarbon Chains During Fischer–Tropsch Synthesis on Ru. *J. Catal.* **2016**, *337*, 91–101.
- (25) Hibbitts, D. D.; Loveless, B. T.; Neurock, M.; Iglesia, E. Mechanistic Role of Water on the Rate and Selectivity of Fischer–Tropsch Synthesis on Ruthenium Catalysts. *Angew. Chem. Int. Ed. Engl.* **2013**, *52*, 12273–12278.

- (26) Hibbitts, D.; Iglesia, E. Prevalence of Bimolecular Routes in the Activation of Diatomic Molecules with Strong Chemical Bonds (O₂, NO, CO, N₂) on Catalytic Surfaces. *Acc. Chem. Res.* **2015**, *48*, 1254–1262.
- (27) Liu, J.; Hibbitts, D.; Iglesia, E. Dense CO Adlayers as Enablers of CO Hydrogenation Turnovers on Ru Surfaces. *J. Am. Chem. Soc.* **2017**, *139*, 11789–11802.
- (28) DeLuca, M.; Kravchenko, P.; Hoffman, A.; Hibbitts, D. D. Mechanism and Kinetics of Methylating C₆–C₁₂ Methylbenzenes with Methanol and DME in H-MFI Zeolites. *In Review, available as pre-print* <http://rxiv.link/8016017>.
- (29) Hibbitts, D. D.; Flaherty, D. W.; Iglesia, E. Effects of Chain Length on the Mechanism and Rates of Metal-catalyzed Hydrogenolysis of N-alkanes. *J. Phys. Chem. C* **2016**, *120*, 8125–8138.
- (30) Hibbitts, D. D.; Flaherty, D. W.; Iglesia, E. Role of Branching on the Rate and Mechanism of C–C Cleavage in Alkanes on Metal Surfaces. *ACS Catal.* **2016**, *6*, 469–482.
- (31) Flaherty, D. W.; Hibbitts, D. D.; Iglesia, E. Metal-catalyzed C–C Bond Cleavage in Alkanes: Effects of Methyl Substitution on Transition-state Structures and Stability. *J. Am. Chem. Soc.* **2014**, *136*, 9664–9676.
- (32) Nystrom, S.; Hoffman, A.; Hibbitts, D. Tuning Brønsted Acid Strength by Altering Site Proximity in CHA Framework Zeolites. *ACS Catal.* **2018**, *8*, 7842–7860.
- (33) Van der Mynsbrugge, J.; Moors, S. L. C.; De Wispelaere, K.; Van Speybroeck, V. Insight into the Formation and Reactivity of Framework-Bound Methoxide Species in H-ZSM-5 from Static and Dynamic Molecular Simulations. *ChemCatChem* **2014**, *6*, 1906–1918.
- (34) Moors, S. L. C.; De Wispelaere, K.; Van der Mynsbrugge, J.; Waroquier, M.; Van Speybroeck, V. Molecular Dynamics Kinetic Study on the Zeolite-Catalyzed Benzene Methylation in ZSM-5. *ACS Catal.* **2013**, *3*, 2556–2567.
- (35) De Wispelaere, K.; Bailleul, S.; Van Speybroeck, V. Towards Molecular Control of Elementary Reactions in Zeolite Catalysis by Advanced Molecular Simulations Mimicking Operating Conditions. *Catal. Sci. Technol.* **2016**, *6*, 2686–2705.