
CLEAVE GUI

Release 0.2.3

A. Tygesen

Mar 17, 2024

CONTENTS:

1 Contact	3
--------------------	----------

This is a graphical user interface (GUI) based on Jupyter notebooks for the Cluster Expansion (CE) code [CLEAVE](#).

The GUI has capabilities of doing several of the standard things you would normally want to use CLEAVE for. However, there are still many features of CLEAVE which are (currently) not implemented in the GUI.

CONTACT

If you encounter any issues or would like to request features, please don't hesitate to reach out on our [GitLab issues](#) page.

1.1 Installation

1.1.1 PyPI

The CLEASE GUI can be installed from [PyPI](#):

```
pip install clease-gui
```

1.1.2 Anaconda

The GUI is also available with anaconda on [conda-forge](#). To install into your conda environment, do:

```
$ conda install -c conda-forge clease-gui
```

Note: This is the preferred method on Windows, due to the dependency on CLEASE. For more information, see [CLEASE docs](#).

1.1.3 Directly from GitLab

Alternatively, it can be installed directly from the master branch via GitLab:

```
pip install git+https://gitlab.com/computationalmaterials/cleas-gui.git@master
```

1.2 Launching the GUI

1.2.1 Quick Start

The GUI is based on Jupyter notebooks. You can get a new notebook to run the GUI in, by running the following:

```
$ clease-gui new
```

This creates a new notebook called `clease_gui.ipynb`. (Alternatively, you can give it a different name with the `-f` flag). Then make a new jupyter lab server,

```
$ jupyter-lab
```

Open the notebook file you just created, and start using the GUI.

1.2.2 Manually Setting up a Notebook

Alternatively, you can manually create a new notebook that you want to work in, and add the following lines in an empty cell:

```
%matplotlib tk  
  
from clease_gui import display_ui  
display_ui();
```

1.3 Examples

1.3.1 Au-Cu Alloy

In this example, we will be looking at the classic [Au-Cu example](#) from the CLEASE documentation, but run with the GUI instead. We will be using a Face-Centered Cubic (FCC) Au-Cu alloy. In you have not used CLEASE before, we recommend that you start by reading the original example from the CLEASE documentation.

First step is to launch the notebook, see [Launching the GUI](#).

Concentration

The first step is the specify the elements we're working on. At the time of writing, it is not possible to specify the concentration ranges, but that is coming soon.

First, specify the basis as `Au Cu`, and click the `Add basis` button, and your concentration pane should look something like this:

Concentration
Settings

Add or remove basis elements to the concentration

Basis:

Add basis

Remove last basis

Clear basis

Current basis: [['Au', 'Cu']]

Settings

Next, we have to specify the Cluster Expansion (CE) settings. Go to the Settings pane. Set the Crystal Structure to FCC, and set a to 3.8. Leave the rest of the settings to their default values. The crystal settings should look like this:

Type:

Basis function:

Crystal Structure:

a:

c:

c/a:

Then, click on the Make Settings above to construct the CE settings. After the Settings have been made, check the status bar in the bottom, which should have a checkmark at the Settings available? indicator. The status indicator should now look like this:

Status: Idle

ECI available? ✗

Settings available? ✔

Supercell available? ✗

Finally, let's save our settings object, so we can use it later. To do that, simply click the Save settings button. If you need to load the settings later, click the Load settings button.

Note: If you ever construct a database, and then later change the settings, (e.g. to adjust the cluster cutoffs), you may need to reconfigure the database. To do this, simply click on the Reconfigure DB button in the settings page.

Creating New Structures

Next, we need to create some structures, so that we can eventually start fitting our CE model.

Go to the **New Structures** tab. Here, we can create new structures with different modes. The default mode is to generate random structures at random concentrations. To get started, let's make 20 new structures, so put 20 in the field **No. struct per gen**, and click the **Generate** button. This will make the 2 end points (pure Au and pure Cu), as well as 18 randomized structures. You can generate more structures by clicking the button again, which will probably become relevant later, if our fit is not satisfactory.

Inspection

We can inspect the structures which are present in our database in the **Inspect DB** tab. Simply click the **Show DB content** to see the list of available structures.

An empty query box means showing all of the structures in the database. For more details on how to construct queries, please see [here](#).

They can be visualized using the **Open in ASE GUI** tab, which will open an external window. For more information on the ASE GUI, please refer to the [ASE GUI documentation](#).

Calculations

Warning: For educational purposes, a tab for doing EMT calculations has been included. This should **not** be used to any production purposes. Instead, results need to be populated into the database manually, such as demonstrated on in the [CLEASE documentation](#).

For demonstration purposes, we will use the built-in EMT calculator in this example. Click the **Calculate with EMT** to run EMT on all the generated structures. And don't worry about clicking the button multiple times, only structures with no "final" counterpart will be run. Once the calculations are finished, they are automatically placed back into the database.

ECI

The effective cluster interactions (ECI) are the coefficients we attribute to any given cluster, in order to predict energies on new structures. We can fit these to our pre-calculated data in the **ECI** tab.

Here, we have two fitting algorithm options: LASSO (also known as ℓ_1) and ℓ_2 . In this case, we will just use ℓ_1 regularization. In the ℓ_1 regularization scheme, we have a single hyperparameter to optimize: α .

In the GUI, this is pretty simple: tick the **Optimize hyperparameter** box, and let's set the **Spacing** type to **Logarithmic Spacing**, min to 1e-8, max to 1, and steps to 100. This optimizes our ECI values with 100 different alpha values in the range [1e-8; 1] with logarithmic spacing, i.e. the sampling density will be lower in the lower ends than in the higher end. Your ECI page should look something along these lines:

Fit ECI's

Fit Plots

Fitting Algorithm: LASSO

☒ Optimize Hyperparameter?

Value: 0.01

Hyperparameter: alpha

Spacing type: Linear Spacing

min: 1e-8

max: 1

steps: 100

Scoring scheme: Leave-one-out CV

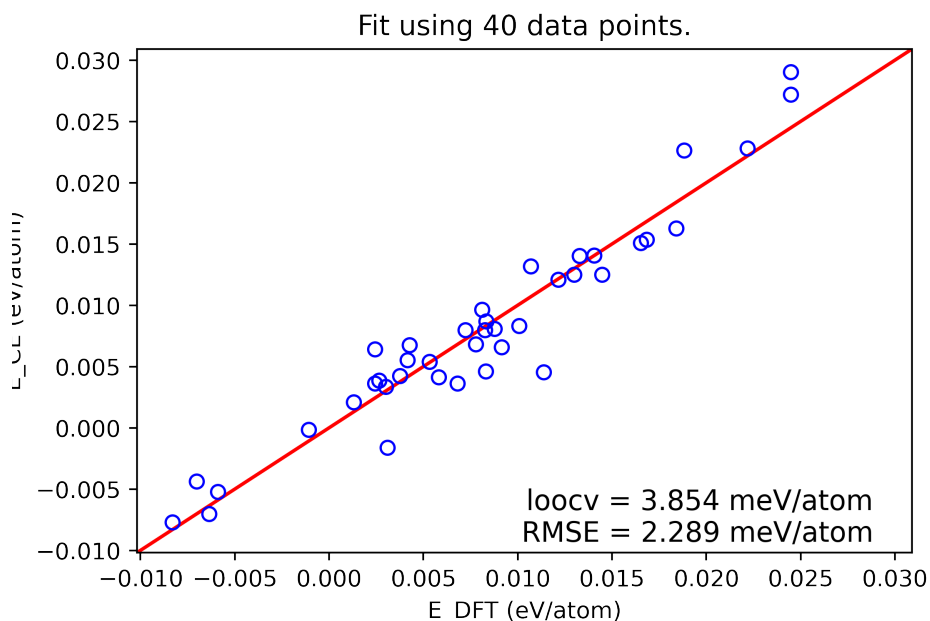
Number of splits: 10

Select a database: Leave empty to use the database specified in the main settings.

Fit ECI

Finally, click on the Fit ECI button to start the hyperparameter optimization. Once the optimization is complete, the status bar in the bottom should have a green check mark by the ECI available? indicator. This means, that the ECI values which best fit our data set has been selected, and is ready to be used for Monte-Carlo sampling.

You can see how good of a fit you got in the Fit Plots tab. For example, here is a correlation plot with 40 structures:



The “DFT” energies (in this case EMT) are the true values we attempt to fit, and are situated along the x -axis. The CE energies are along the y -axis. Ideally, these points should be as close to the $x = y$ line as possible, but that is determined by how good fits we are able to obtain.

Supercells

One of the nice things about cluster expansion, is that it allows us to perform calculations on supercells we would normally not be able to run with regular DFT, especially when combined with techniques such as Monte Carlo sampling.

Go to the Supercell tab, and select the size of the cell. The repetitions in the (x, y, z) directions are relative to the primitive cell. Let's make a (6x6x6) supercell. This gives us a new atoms object with 216 atoms. The page should, besides the specific ratio of Au and Cu atoms, looks like this:

Create a supercell. Current formula unit: Au198Cu18. Total number of atoms: 216

Save Supercell	Filename: <input type="text" value="supercell.traj"/>
Load Supercell	Filename: <input type="text" value="supercell.traj"/>
Make random supercell	View Current Supercell
Repetition (x):	<input type="text" value="6"/>
Repetition (y):	<input type="text" value="6"/>
Repetition (z):	<input type="text" value="6"/>

We can now use the bottom panel to specify a ratio of Au and Cu atoms. Let's do the 50/50 cell, i.e. 108 Au and 108 Cu atoms.

Sublattice 1

Au:

Cu:

Total specified: Total in basis:

Update supercell

Finally, click the Update Supercell button to set the number of Au and Cu atoms. You should now have a green check mark in the Supercell available? indicator in the status bar.

Monte Carlo

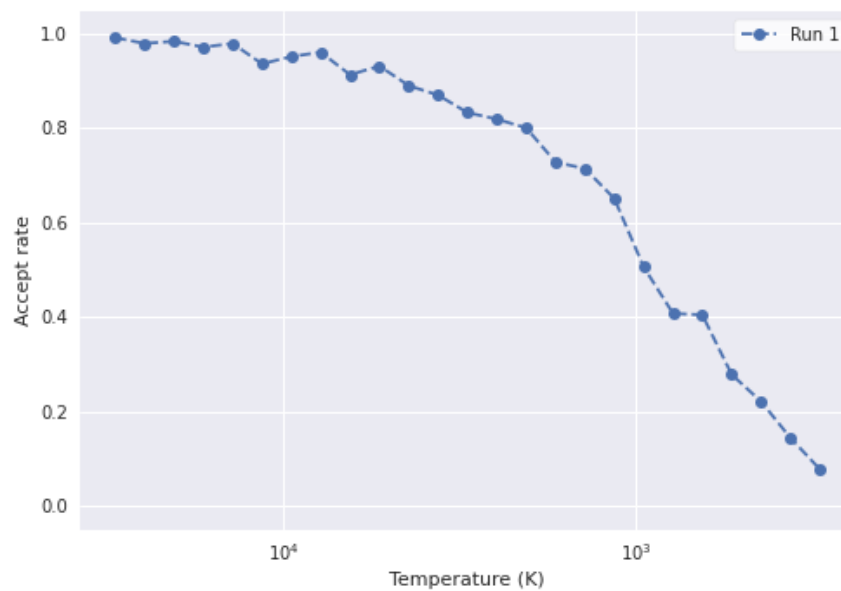
Note: Presently, the GUI only supports regular Monte Carlo sampling.

With settings, ECI and supercell available, we can proceed to performing a Monte-Carlo (MC) sampling of the supercell we just made.

In the **Monte Carlo** tab, we see a few options. The first is the **Step** mode, where we have the options *sweeps* or *steps*. 1 sweep is defined as running N number of steps, where N is the number of atoms in the supercell. Let's leave it on *sweeps*.

Next, we set the number of sweeps, let's set that to 3. That means, in this case, we'll run $3 * 216 = 648$ steps in our Monte Carlo simulation. Finally, let's increase the number of temperature steps to 100. Once you're satisfied with your MC settings, click the **Run MC** button.

We can use the **Plotting** tab to visualize the run. A good indicator of whether the MC sampling was sufficient, is the **Accept Rate**, which should be a smooth downwards curve, which eventually flattens at close to 0.0. In this case, we found an accept rate that looked something like:

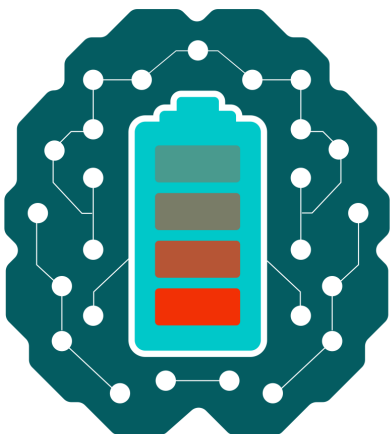


This curve is clearly not very smooth, which indicates that the number of iterations we ran was insufficient to reach convergence. You can go back and re-run the MC with more steps, to further improve your convergence.

To see the current state of the supercell, go back to the **Supercell** tab and click on the **View current supercell** button, which will display the latest configuration which was found in the MC run. You can export the supercell into a file with the **Save Supercell** button.

1.4 Partners and Support

Development of the CLEASE GUI was supported by [BIG-MAP](#) (H2020, #957189).



BIG MAP