

## Table of Contents

Table of Contents	1
Crystal Structure Prediction Scripter: Phases of TiO <sub>2</sub>	2
Setting up the calculation	2
Crystal Structure Prediction Scripter	3
Genetic algorithm settings	3
Adding a calculator	6
Running the calculations and analyzing results	7
References	8

[Docs](#) » [Tutorials](#) » [Materials, surfaces and chemistry](#) »Crystal Structure Prediction Scripter: Phases of  $\text{TiO}_2$ 

## Crystal Structure Prediction Scripter: Phases of $\text{TiO}_2$

Version: 2016.2

### Downloads & Links

[PDF version](#) [Slides](#)[Basic QuantumATK Tutorial](#)[ATK Reference Manual](#)

In this tutorial, we will use the Crystal Structure Prediction method to investigate the possible crystal structures of  $\text{TiO}_2$ . At zero temperature and pressure, the *rutile* phase of  $\text{TiO}_2$  is the most stable one, while the *anatase* and *brookite* phases are meta-stable. Both rutile and anatase can be described in a unit cell of 6 atoms, while brookite requires a much larger cell. QuantumATK uses a genetic algorithm for crystal structure prediction.

### Note

A **genetic algorithm** is a method for solving both constrained and unconstrained optimization problems based on a “natural selection” process that mimics biological evolution. The algorithm repeatedly modifies/refines a population of individual solutions.

In practice, the algorithm starts out from an initial population of crystals (a generation), and iteratively refines the generation by producing new generations through application of **genetic operators** to the previous generation, e.g. *promotion* (lowest-energy crystals live on), *heredity* (mixing of “good genes” from two different low-energy crystals), as well as *mutation* and *permutation* (ensures genetic diversity).

Please refer to [\[1\]](#), [\[2\]](#) and [\[3\]](#) for more information.



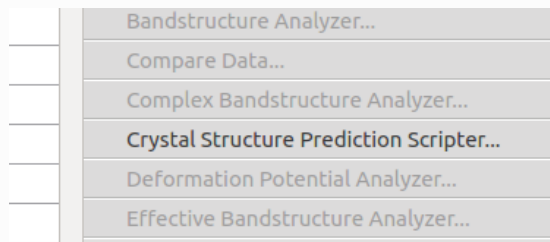
## Setting up the calculation

We will use the **Crystal Structure Prediction Scripter**, which is available in recent versions of QuantumATK.

This custom scripter widget is used to produce a template script for running the crystal structure prediction algorithm. The scripter does not add script lines defining the specific calculator engine to be used – they must be added manually.

## Crystal Structure Prediction Scripter

Open the Crystal Structure Prediction Scripter from the right-hand plugins panel in QuantumATK.



The widget will open, and will initially display the “I/O” tab.

First of all, you need to write the **chemical formula** of the crystal to investigate. In this case, you should write “(TiO2)2” (or “Ti2O4”) as we need two TiO<sub>2</sub> formula units to describe the unit cells of both anatase and rutile. Leave the external pressure at 0 GPa. The initial volume will be set automatically, based on the expected volume of the elements. The window should now look as seen below.

A screenshot of the Crystal Structure Prediction Scripter window. The 'File' menu is open. The 'Chemical Formula' field contains '(TiO2)2'. Below it, a message says 'There will be 6 atoms per unit cell.' The 'External pressure' field is set to '0' GPa. The 'Initial volume' field is set to '51.57' Å³. There are three tabs: 'Genetic Algorithm', 'Optimization', and 'I/O'. The 'I/O' tab is selected. Under the 'I/O' tab, there are fields for 'Random number seed' (set to 'Automatic'), 'Initial population log prefix' (set to 'initial\_population\_'), and 'Crystal structure prediction log prefix' (set to 'generation\_'). There is also a checked checkbox for 'Write population'. A blue arrow button is at the bottom right.

## Genetic algorithm settings

Next, set up the parameters for the genetic algorithm, which is used to search for viable crystal structures. Click the tab named “Genetic Algorithm” to start. It has the following parameters:

### Population size:

The number of crystal structures in each generation. Set this to 10.

#### Number of generations:

The number of steps the algorithm will run. Set this to 20.

#### Selection pressure:

Determines how strongly the fitness score (see note below) will be taken into account when individuals are selected for application of a genetic operator. A high value will make it much more likely that genetic operators are applied to the individuals with the highest fitness, while a value of 0 will make all (elite, see below) individuals equally likely. Leave this value at 2.00.

#### Number of elites:

The number of individuals which will be considered when creating the next generation. The difference between "Population size" and "Number of elites" indicates the number of individuals with lowest fitness that will simply be discarded after each iteration step. Leave this value at 10, so that no individuals will be discarded.

#### Number to promote:

The number of individuals, with the highest fitness scores, which are promoted (copied) to the next step without any alteration. Set this to 4.

#### Note

The fitness score is defined as minus the enthalpy. This ensures that the algorithm searches for the thermodynamically most stable structures.

The remaining members of each new generation are created by applying **genetic operators** to the elite individuals from the previous generation. There are three genetic operators, with an additional setting for one of them. The probability of applying each operator is calculated from the following **weights**, which may be considered to be unnormalized probabilities.

#### Heredity weight:

The unnormalized probability that the *Heredity* operation is used to create the next individual in the next generation. This operator uses two structures, and combines one-half of each as determined by a randomly chosen cut-plane. Set this value to 50.

#### Permutation weight:

The unnormalized probability of using the *Permutation* operator. This operator will exchange the positions of two randomly selected atoms of different elements. Set this value to 20.

#### Mutation weight:

The unnormalized probability of using the *Mutation* operator. This operator applies a symmetric strain matrix with elements drawn randomly from a gaussian distribution with a standard deviation of *Mutation lattice strain*. This will change the lattice parameters and angles of the unit cell. Set the *Mutation weight* to 30 and leave the *Mutation lattice strain* at 0.7.

The Crystal Structure Prediction Scripter should now look like this:

File

Chemical Formula

There will be 6 atoms per unit cell.

External pressure  GPa

Initial volume  Å<sup>3</sup>

Genetic Algorithm Optimization I/O

Population Size

Number of generations

Selection pressure

Number of elites

Number to promote

**Genetic Operators**

Heredity weight

Permutation weight

Mutation weight

Mutation lattice strain



Next, go to the **Optimization** tab and verify that it looks like this:

File

Chemical Formula

There will be 6 atoms per unit cell.

External pressure  GPa

Initial volume  Å<sup>3</sup>

Genetic Algorithm Optimization I/O

Force tolerance  eV/Å

Pressure tolerance  GPa



Maximum number of steps

Maximum step length  Å




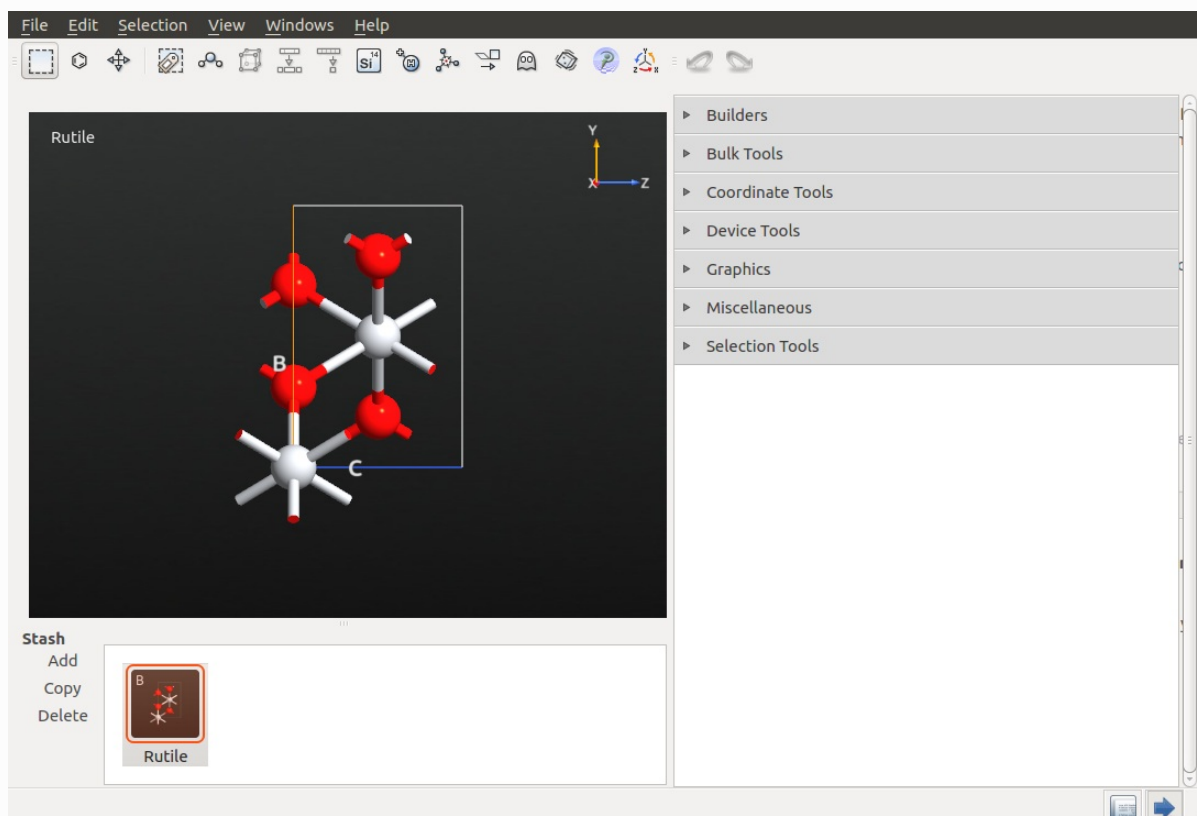
### Tip


You can of course change the geometry optimization settings, e.g. to tighten the force tolerance, but the default values should work well for most purposes.

Send the script to the  **Editor** using the  button. The produced script should be identical to this one: [csp-script.py](#). Note that `calculator = None`. You therefore need to manually add the script lines defining the calculator to be used (could be an *ATK-DFT* or *ATK-ForceField* calculator).

## Adding a calculator

In principle, you can just add a calculator to the script, particularly if you already have an appropriate one defined in another script or can write it up manually. However, here we will go through the steps of defining it using QuantumATK. Open the  **Builder** and find the rutile structure of  $\text{TiO}_2$ :



Send this configuration to the  **Script Generator** and set up an **ATK-ForceField** calculator. QuantumATK will automatically suggest relevant parameter sets. In this case, choose **Pedone\_2006Fe2**.

### Note

For QuantumATK-versions older than 2017, the **ATK-ForceField** calculator can be found under the name **ATK-Classical**.

**New Calculator**

**Calculators**

- ☐ ATK-DFT: LCAO
- ☐ ATK-DFT: Plane-wave (beta)
- ☐ ATK-SE: Extended Hückel
- ☐ ATK-SE: Slater-Koster
- ☒ ATK-ForceField
- ☐ FHI-aims

**Potential Settings**

Parameter set:

**Literature References**

A. Pedone and G. Malavasi and M. Menziani and A. Cormack and U. Segre, *A new self-consistent empirical interatomic potential model for oxides, silicates and silica-based glasses*, J. Phys. Chem. B, 110, pp. 11780-11795, 2006.

**Additional Info**

**IO**

☒ Save ☒ Print

File:  ... Label:

Send the script to the **Editor**, and copy the script lines defining the calculator:

```
potentialSet = Pedone_2006Fe2()
calculator = TremoloXCalculator(parameters=potentialSet)
calculator.setVerletListsDelta(0.25*Angstrom)
```

Insert those lines into the script generated using the Crystal Structure Prediction Scripter, and save it. You can also download the complete script here: [csp-script-with-calculator.py](#)

## Running the calculations and analyzing results

Run the script using the **Job manager** or from the command line. It will take no more than a few minutes to run.

The script will generate a significant number of files, since it saves an `.nc` file and 6 log-files for each generation. The `.nc` files contain the *BulkConfigurations* for each individual in the given generation, and the log-files contain information from the 6 geometry optimizations carried out in each generation. This is the total population minus the number of promoted individuals, as they do not need to be re-optimized. However, we will focus on the overall log-file, in this case called `csp-script-with-calculator.log`. We are particularly interested in the information about the final generation:

```

+-----+
| Niching                                     |
+-----+
| Killing individual  0                      |
| Killing individual  1                      |
| Killing individual  4                      |
| Killing individual  6                      |
+-----+
| Population for Final Niche Generation    |
+-----+
| Individual  0: Fitness:  94.238940         |
|           Symmetry: P4_2/mnm      Volume:  63.5550         |
|           Max Force: 0.0056      Max Stress Error: 0.0929   |
|           History: IPPPPPPPPPPPPPPPPPPPP |
| Individual  1: Fitness:  93.978255         |
|           Symmetry: I4_1/amd      Volume:  70.0359         |
|           Max Force: 0.0025      Max Stress Error: 0.0380   |
|           History: IPPPPPPPPPPPPPPPPPPPP |
| Individual  2: Fitness:  93.796305         |
|           Symmetry: Fd-3m         Volume:  139.6049        |
|           Max Force: 0.0099      Max Stress Error: 0.0357   |
|           History: IXPPPPPPPPPPPPPPPPPPPP |
| Individual  3: Fitness:  93.625090         |
|           Symmetry: Cmcmm         Volume:  87.9469         |
|           Max Force: 0.0074      Max Stress Error: 0.0533   |
|           History: IPPPPPPPPPPPPPPPPPPPP |
| Individual  4: Fitness:  93.388660         |
|           Symmetry: Imcm          Volume:  100.2970        |
|           Max Force: 0.0092      Max Stress Error: 0.0912   |
|           History: IPPPPPPPPPPPPPPPPPPPH |
| Individual  5: Fitness:  92.364891         |
|           Symmetry: Imm2          Volume:  86.5277         |
|           Max Force: 0.0042      Max Stress Error: 0.0761   |
|           History: IPPPPPPPPPPPPPPPPPPPH |
+-----+
| Legend: I=Initial P=Promotion H=Heredity M=Mutation X=Permutation |
|           fitness in units of eV                                     |
|           force in units of eV/Angstrom                           |
|           stress in units of GPa                                   |
+-----+

```

First, the genetic algorithm performs **Niching**, which is removal of any duplicate structures from the generation. Below that, we see the final generation, sorted by decreasing fitness, which is simply the negative of the enthalpy. We see that the first two structures (those with lowest energy) have *P4<sub>2</sub>/mnm* and *I4<sub>1</sub>/amd* symmetries, consistent with the rutile and anatase phases of TiO<sub>2</sub>.

The **History** shows how the algorithm arrived at this particular generation, and it turns out that, in this case, both the rutile and anatase structures have simply been promoted all the way from the initial population. Note also how the volume varies quite a bit between the different crystal structures, showing how the genetic algorithm is able to find highly different local minima on the potential energy surface of this system.

## References

[1]

William W Tipton and Richard G Hennig. A grand canonical genetic algorithm for the prediction of multi-component phase diagrams and testing of empirical potentials. *J. Phys. Condens. Matter*, 25(49):495401, 2013. doi:10.1088/0953-8984/25/49/495401.

[2]


Artem R. Oganov. *Modern Methods of Crystal Structure Prediction*. Wiley, 2011. ISBN 9783527643776.


[3]

Eva Zurek. *Discovering New Materials via A Priori Crystal Structure Prediction*, pages 274–326. John Wiley



& Sons, Inc, 2016. [doi:10.1002/9781119148739.ch5](https://doi.org/10.1002/9781119148739.ch5).

 Previous

Next 

---

© Copyright 2023 Synopsys, Inc. All Rights Reserved.