

# Silico

Computational Chemistry Management

*Quick Reference Guide*

Version 1.0.0

14 February 2022

Oliver S. Lee, Campbell F. R. Mackenzie, Tomas Matulaitis, Ettore Crovini, Eli Zysman-Colman

# Table of Contents

<b>How to Use This Document.....</b>	<b>4</b>
<b>About Silico.....</b>	<b>5</b>
<b>Installation.....</b>	<b>6</b>
<b>Introduction.....</b>	<b>7</b>
The Sub-programs.....	7
Running interactively.....	7
<b>Preparing To Submit Calculations.....</b>	<b>9</b>
Prepare Files.....	9
Upload Files.....	9
Universal Input Format (.si).....	9
<b>Submitting Calculations.....</b>	<b>11</b>
<b>Interactively.....</b>	<b>11</b>
Submission Interface.....	12
Selecting Coordinates.....	12
Selecting Methods.....	12
Destination.....	13
Program.....	14
Calculation.....	15
Confirm Selection.....	17
<b>Method Codes and Non-Interactive Submission.....</b>	<b>18</b>
Custom Method Files.....	18
Common Codes.....	19
<b>Turbomole SCS-CC2 and MR-TADF.....</b>	<b>20</b>
SCS-ADC(2).....	21
<b>Folder Structure.....</b>	<b>23</b>
Input.....	23
Output.....	23
Flags.....	23
Results.....	23
Report.....	23
<b>Monitoring Calculations.....</b>	<b>24</b>
The Flags Folder.....	24
List of Flags.....	25
Multiple Calculations.....	26
<b>Post Analysis.....</b>	<b>27</b>
Results.....	27
Available Result Files.....	28
Report.....	29
Spin-Orbit Coupling (SOC).....	30
<b>Analysis of Results.....</b>	<b>31</b>
Generating Summaries.....	31
Analysis of Multiple Results.....	31
<b>Generating Reports.....</b>	<b>33</b>
Normal Usage.....	33
Combined Reports.....	33
Automatic Combined Reports.....	34
<b>Checking The Queue.....</b>	<b>35</b>
<b>Referencing Silico.....</b>	<b>36</b>
Calculation submission.....	36
Calculation results.....	36

## Silico 1.0.0 – Computational Chemistry Management – Quick Reference Guide

Molecular and orbital images.....	36
Graphs.....	36
Spin-orbit coupling.....	36
<b>Example Reference.....</b>	<b>36</b>
<b>Bibliography.....</b>	<b>37</b>

## How to Use This Document

This document is designed to act as rapid introduction and quick-reference to the Silico Computational Chemistry Management system.

Images are included for comparison and reference.

Supplementary information and helpful tips are displayed separately, as follows:

**NOTE:** This is a tip.

Commands that should be typed by the user are displayed in the following format:

```
> echo Hello world
```

The arrow character (>) should not be typed; it is used to differentiate commands that should be typed from the resulting output (which will be displayed without the arrow character):

```
> echo Hello world  
Hello world
```

In the above example, the user is being instructed to type: echo Hello world. The computer, in response, gives the output: Hello world.

Ellipses (...) indicate that the real, full output has been truncated:

```
> cat /etc/fstab  
# /etc/fstab: static file system information.  
#  
...
```

## About Silico

Computational chemistry, particularly for newcomers, can be an almost impenetrably complex field. Silico is a software package that attempts to alleviate this problem by automating and/or simplifying as many aspects of computational chemistry as possible, so that the computational process becomes as close to a black-box as can be achieved.

Silico is under development and suggestions are welcome, please contact [osl@st-andrews.ac.uk](mailto:osl@st-andrews.ac.uk).

A non-exhaustive list of features is as follows:

- Submission to computational programs through a simple and unified interface.
- Simultaneous submission of multiple molecules/systems.
- Automatic in series submission of results from completed calculations to subsequent calculations.
- Automatic conversion of input files (including ChemDraw) to types appropriate for the selected computational program.
- Automatic and manual analysis of computation results, including tabulation to CSV format.
- Automatic and manual generation of PDF reports from computation results, including rendered 3D structures, orbital images and graphs.

# Installation

To install Silico, connect to Kennedy using PuTTY and log-in to your account as normal. Once logged in, use the following command:

```
> /gpfs1/apps/EZC-tools/install-silico
```

**NOTE:** Installation only ever needs to be performed once (for each user). Updates to Silico will be become available automatically without the need to reinstall.

Installation should be near instantaneous. Once complete, a message will be printed:

```
> /gpfs1/apps/EZC-tools/install-silico
Installed successfully. Please log out and in again to complete
```

Use the 'exit' command to log out:

```
> /gpfs1/apps/EZC-tools/install-silico
Installed successfully. Please log out and in again to complete
> exit
logout
```

Once logged back in again, Silico will now be available for use. The installation can be tested by using the 'silico' command with the '-v' (version) option. If successful, the version will be printed:

```
> silico -v
1.0.0
```

If the installation was not successfully, the following error message will be printed:

```
> silico -v
-bash: silico: command not found
```

If unsuccessful, seek help from another group member.

# Introduction

## The Sub-programs

The various functionality of Silico is split into a number of sub-programs. These can be accessed by specifying the name of the desired sub-program after the `silico` command. For example, to use the submit program:

```
> silico submit
```

Each sub-program also has a 3-letter and 1-letter short code that can be used to access the program. There is no difference between using the short codes or the full name, so the user is encouraged to use whichever style they prefer. For the purposes of this document, all examples will use the 3-letter short code (unless otherwise stated). As example, the submit program can also be accessed by any of the following commands:

```
> silico sub
```

or:

```
> silico s
```

A list of the available sub-programs is given below:

Command	3-letter	1-letter	Description
<code>silico submit</code>	sub	s	Submit calculations
<code>silico report</code>	rep	r	Generate reports from completed calculations
<code>silico result</code>	res	R	Analyse and tabulate calculation results
<code>silico convert</code>	con	c	Convert different input/output file formats
<code>silico status</code>	sta	S	Check how busy the queue is
<code>silico interactive</code>	int	i	Run silico interactively

## Running interactively

All of the Silico sub-programs (except '`silico status`') have both a text/console interface as well as an interactive interface. By default, each sub-program will use the text interface which is generally faster for experienced users and can be easily incorporated into custom scripts. However, many users will find the interactive interface easier to use and generally more forgiving. To use the interactive interface, specify the '`-I`' (or '`--interactive`') option after the sub-program.

For example, to use the submit program interactively:

```
> silico submit -I
```

or

## Silico 1.0.0 – Computational Chemistry Management – Quick Reference Guide

```
> silico submit -I
```

or:

```
> silico sub -I
```

or:

```
> silico s -I
```

In addition, the 'silico interactive' command can be used to launch the Silico interactive main menu directly:

```
> silico interactive
```

or:

```
> silico int
```

or:

```
> silico i
```



From here, the interactive interfaces to the various sub-programs can be chosen directly.



# Preparing To Submit Calculations

## Prepare Files

To submit a calculation, the system(s) of interest first need to be prepared by drawing the relevant structures electronically (on a program). Silico supports a wide variety of input file types, including both 2D and 3D formats. Notable entries include:

- G09 output (.log)
- GaussView (.com, .gjf, .gjc and .gau)
- ChemDraw (.cdx and .cdxml)
- Avogadro (.cml)
- MarvinSketch (.mrv)
- Crystallographic Information File (.cif)

**NOTE:** MarvinSketch (.cml) and Avogadro (.mrv) are free alternatives to ChemDraw and GaussView, respectively.

Silico uses the OpenBabel library for file conversion. Please see <https://open-babel.readthedocs.io/en/latest/FileFormats/Overview.html> for a full list of supported formats.

**NOTE:** When using ChemDraw files (.cdx), remember to save each molecule individually to its own file. If a single file contains multiple structures, then all these structures will be submitted together to a single calculation.

In general, the closer to the final atom coordinates the input coordinates are, the fewer optimisation cycles will be required. As such, it is strongly recommended to favour 3D formats over 2D. If available, a Crystallographic Information File (.cif) or output file from a previous calculation (on the same or similar structure) should be favoured as input, for the same reason.

## Upload Files

Once the structures have been prepared, transfer the input files to Kennedy using the WinSCP program. It is recommended that input files for related calculations be stored inside subfolders within your home directory on Kennedy.

## Universal Input Format (.si)

Silico supports a text-based, program-independent input format simply called the silico input format (.si). This format is useful because the input formats native to certain computational programs, notably Turbomole, do not contain molecular charge or multiplicity information which is required for a calculation. This information can be set in the .si format.

To create a .si file, use the 'silico convert' command to convert any file format supported by Silico:

## Silico 1.0.0 – Computational Chemistry Management – Quick Reference Guide

```
> silico con Benzene.com -O Benzene.si
```

The new file will be written to 'Benzene.si', and will have the following structure:

```
name: Benzene
charge: 0
multiplicity: 1
geometry: |-
  C      -1.20910      -0.69809      0.00000
  C      -1.20908      0.69806      0.00000
  C       0.00000     -1.39611      0.00000
  C       1.20909     -0.69804      0.00000
  C      -0.00000      1.39611      0.00000
  C       1.20910      0.69807      0.00000
  H      -2.14965     -1.24108      0.00000
  H      -0.00001     -2.48216      0.00000
  H      -2.14967      1.24098      0.00000
  H       0.00004      2.48216      0.00000
  H       2.14961      1.24116      0.00000
  H       2.14962     -1.24109      0.00000
```

This file can then be edited by the user, for example with the 'nano' program, to change the desired name, charge and/or multiplicity. The input file can then be submitted as normal.

# Submitting Calculations

## Interactively

Calculations can be submitted using the 'silico sub -I' command, which can optionally be followed by a list of coordinate files to submit. For example, to submit a file called 'Benzene.com', the following command would be used:

```
> silico sub -I Benzene.com
```

An explicit charge and multiplicity can be set by using the '-C' (**charge**) and '-M' (**multiplicity**) options. If given, these options will override any charge or multiplicity information in the input file.

```
> silico sub -I Benzene.com -M 3 -C 0
```

If the file name contains spaces, remember to use speech marks:

```
> silico sub -I "Methyl benzene.com"
```

If the file is not in the current directory, specify the full path to the input file (including directories):

```
> silico sub -I "Aromatic Folder/Methyl benzene.com"
```

**NOTE:** You can check what files are in the current directory by using the 'ls' command.

Alternatively, you can use the 'cd' command to change into the subdirectory before using Silico:

```
> cd "Aromatic Folder"
> silico sub -I "Methyl benzene.com"
```

One of the main advantages of using Silico for calculation submission is the ability to submit multiple calculations at once. This can be achieved by separating each file name with a space:

```
> silico Benzene.com Naphthalene.xyz
```

There is no limit on the number of coordinate files that can be submitted at once (beyond limitations of the operating system and memory). In many cases, it is useful to submit the contents of whole directories at once, this can be achieved using the wildcard character (\*). For example, to submit all files in the current directory ending in '.com':

```
> silico sub -I *.com
```

Or to submit all files in another directory:

```
> silico sub -I "Aromatic Folder"/*.com
```

By default, a new folder will be created for each submitted coordinate file in the current folder. The location of these output folders can be changed if desired with the '-O' (**output**) option, which is often useful when submitting coordinates from another folder. For example, to create the new folders in the same directory as above, the command would be:

## Silico 1.0.0 – Computational Chemistry Management – Quick Reference Guide

```
> silico sub -I "Aromatic Folder"/*.com -o "Aromatic Folder"
```

**NOTE:** All files submitted with the same silico command will be submitted to the same calculation. To submit to different calculations, use the silico command multiple times.

If the desired calculation codes are known at this point, they can also be given with the '-c' (**code**) option. For more information on calculation codes, see: Method Codes and Non-Interactive Submission. For example:

```
> silico sub -I Benzene.com -c 1/1/1
```

Multiple codes can be given in the same way as multiple coordinate files:

```
> silico sub -I Benzene.com -c 1/1/1 1/1/2
```

### Submission Interface

The 'silico sub -I' command will launch the interactive submission interface, which appears as follows:

Silico 1.0.0-pre.6 Silico Calculation Submitter					
<b>Input Coordinates</b>					
1	Benzene	C6H6	charge:0	mult:1	< ↑ > ↓ >
2	Naphthalene	C10H8	charge:0	mult:1	< ↑ > ↓ >
3	< Add new here				> < ↑ > ↓ >
<b>Calculation Methods</b>					
1	< Browse library > < Add from code > < Add from files >				< ↑ > ↓ >
<b>Output Options</b>					
Output location: < . >					
< Back >> Confirm					
Arrow Keys: Navigate TAB: Next: SHIFT-TAB: Previous ENTER: Select ESC: Back					

### Selecting Coordinates

The 'Input Coordinates' section will display any loaded coordinate files, along with the molecular formula, charge and multiplicity (the latter two of which can be edited). Additional coordinate files can be loaded using the < Add new here > button, which will open a file browser. Unwanted coordinate files can be removed using the < r > (**r**emove) button for that row.

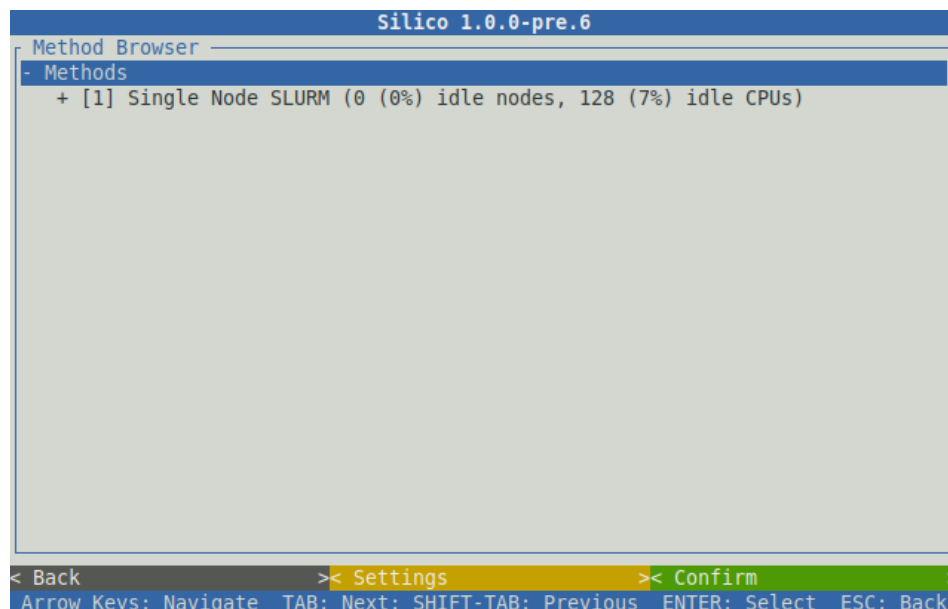
### Selecting Methods

Calculations in Silico are represented by *methods*, each of which contains three individual parts: (1) The Destination: The location to where the calculation will be submitted, for example part of a server cluster, (2) The Program: The computational chemistry program, such as Turbomole or

## Silico 1.0.0 – Computational Chemistry Management – Quick Reference Guide

Gaussian, and (3) The Calculation: The specific calculation to perform, containing information such as the functional and basis set.

Methods can be selected by using the < Browse library > button under the 'Calculation Methods' section, which will open the method browser:



### Destination

The first item to choose is the destination, each of which represents a different group of computing nodes (called partitions). Three are available at the time of writing, they are:

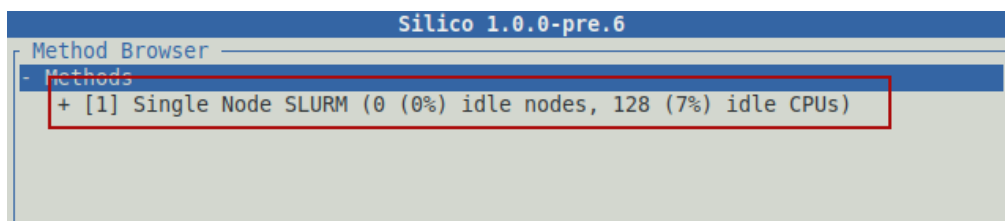
- **Singlenode SLURM:** The default partition, **this should be chosen in nearly all cases.**
- **Debug SLURM:** A partition for testing purposes only. Debug has reduced resources and a time limit; it is not suitable for performing calculations.
- **Multi Node SLURM:** A partition intended for more intensive calculations that require multiple nodes. **Use of the multi-node partition is generally forbidden, you must speak to a senior group member before selecting this partition.**

**NOTE:** Although BigMem SLURM does have more memory available than Singlenode SLURM, the latter already offers > 100 GB per node. This is likely to be more than sufficient for most cases.

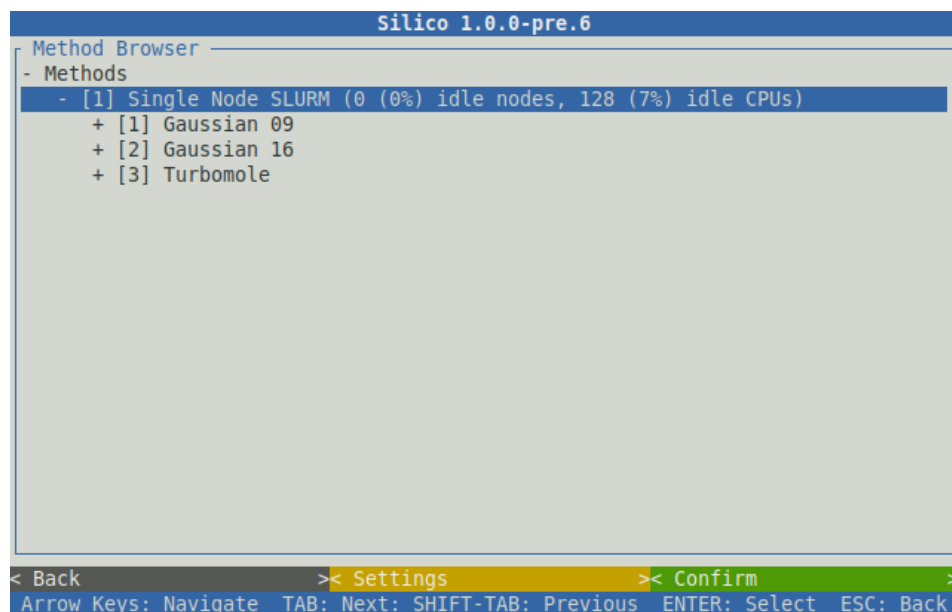
**NOTE:** Debug SLURM and Multi Node SLURM are hidden by default, but they can be revealed by changing the 'show\_hidden' option under 'Settings'.

Next to the name of each method, the status of the relevant partition is printed. This gives an indication of how busy each is at the current moment:

## Silico 1.0.0 – Computational Chemistry Management – Quick Reference Guide



To select a destination, navigate up and/or down using the arrow keys and use 'right arrow' to expand the selection. For example, to select the first method, 'Single Node SLURM':

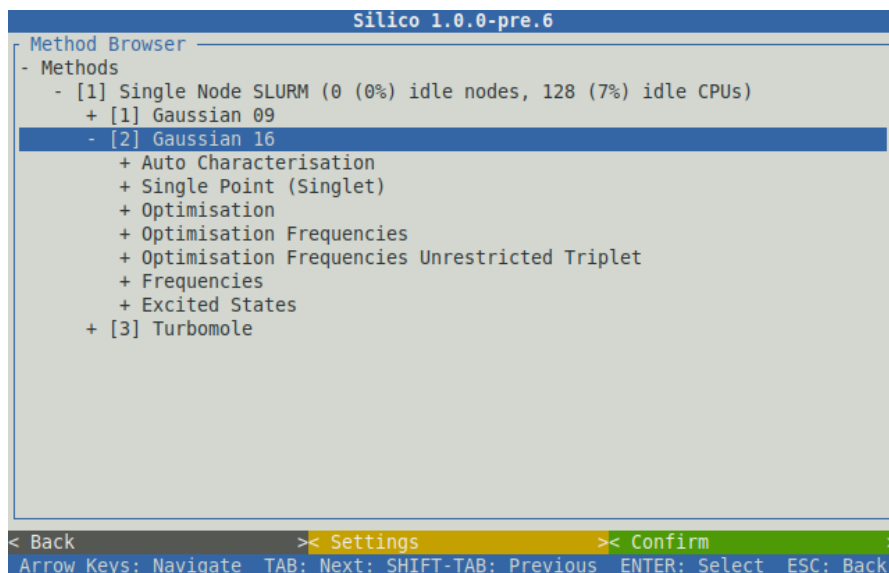


### Program

The next item to choose from the browser is the calculation program. Currently, three calculation programs are supported, they are: Gaussian 09, Gaussian 16 and Turbomole. In general, Gaussian 16 should be chosen for the characterisation of all organic and organometallic emitters and photocatalysts, as this later version offers a number of improvements over the older Gaussian 09. However, it is important to note that calculations with different programs (eg, Gaussian 09 and Gaussian 16) **are not comparable**, and so calculations on a series of molecules must be performed using the same program. Thus if some calculations on a series have already been performed using G09, all further computations on that series should also use G09, or all calculations should be redone using G16.

Turbomole is primarily used for the characterisation of multi-resonant thermally activated delayed fluorescence (MR-TADF) emitters, as the photophysics of these compounds require higher order methods, such as SCS-CC2 or SCS-ADC(2), which are only supported by Turbomole (see Preparing To Submit Calculations/Turbomole SCS-CC2 and MR-TADF) . If your molecule is not a MR-TADF type emitter, you likely to do not need to use Turbomole.

**NOTE:** Special permissions are required to perform calculations with Turbomole. If you do not have access to Turbomole and believe you require these permissions, please contact Oli (osl@st-andrews.ac.uk).



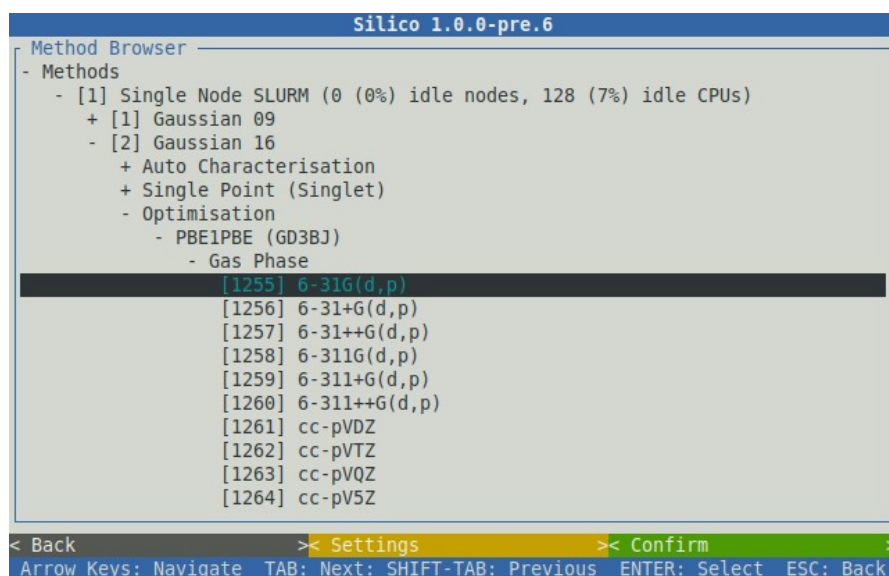
If in doubt, Gaussian 16 is likely the correct program to use. To choose a program, navigate down with the arrow keys and use 'right arrow' to expand your selection:

## Calculation

The final item to choose is the calculation itself. The available calculations are sorted into categories with self descriptive names, each category can be expanded in the same manner as before; with the arrow keys. The categories are as follows:

- Calculation type (Opt, Freq, SP, TD-DFT etc)
- Functional or method (B3LYP, MP2, CCD etc)
- Solvent (only for DFT) or gas phase
- Basis set

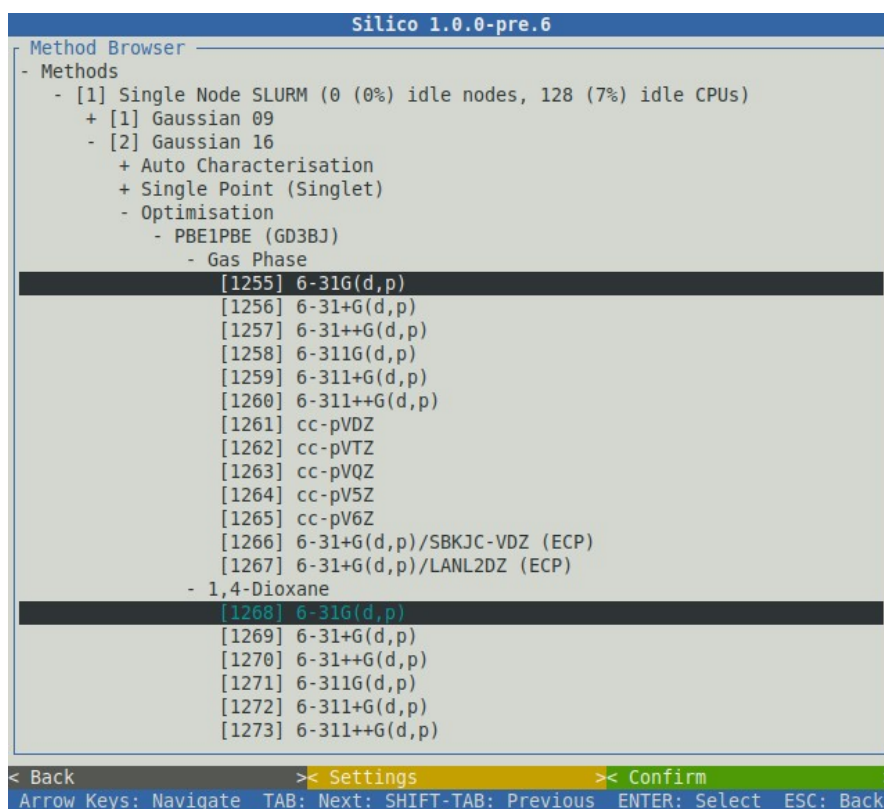
Once the desired calculation has been chosen, pressing enter will select it:



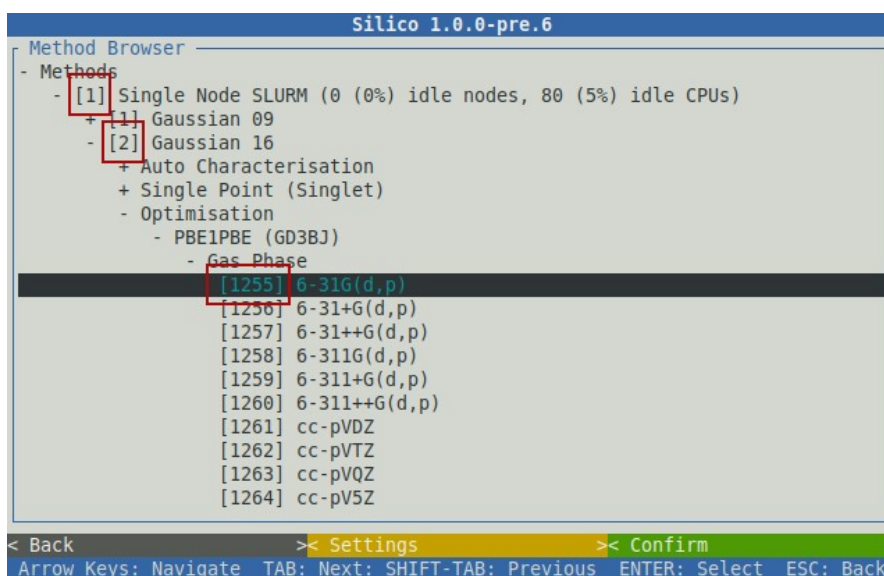
If a calculation is chosen in error, pressing enter again will unselect it.

## Silico 1.0.0 – Computational Chemistry Management – Quick Reference Guide

Multiple calculations can be chosen in the same way and each will be selected in turn:



Each method chosen from the browser can be represented by a unique code of three numbers, each of which identifies the three separate parts that make up that method. For example, the method chosen below has the code 1/2/1255:



Method codes can be used to quickly select a method without having to navigate through the browser (see Method Codes and Non-Interactive Submission). Once a (number of) methods have been chosen, selecting the 'Confirm' button will close the browser and add the selected methods to the queue under the 'Calculation Methods' section:



## Silico 1.0.0 – Computational Chemistry Management – Quick Reference Guide

Silico 1.0.0-pre.6					
Silico Calculation Submitter					
Input Coordinates					
1	Benzene	C6H6	charge:0	mult:1	< ↑ > < ↓ >
					< r >
2	Naphthalene	C10H8	charge:0	mult:1	< ↑ > < ↓ >
					< r >
3	< Add new here				> < ↑ > < ↓ >
					< r >
Calculation Methods					
1	[1/2/1255] Single Node SLURM : Gaussian 16 : Optimisation				< ↑ > < ↓ >
	PBE1PBE (GD3BJ) Gas Phase 6-31G(d,p)				< m > < r >
2	[1/2/6247] Single Node SLURM : Gaussian 16 : Excited States TDA				< ↑ > < ↓ >
	10 Singlets 10 Triplets PBE1PBE (GD3BJ) Gas Phase				< m > < r >
	6-31G(d,p)				
3	< Browse library > < Add from code > < Add from files >				< ↑ > < ↓ >
Output Options					
Output location: < . >					
< Back >> Confirm					
Arrow Keys: Navigate TAB: Next: SHIFT-TAB: Previous ENTER: Select ESC: Back					

Multiple methods that are queued in this way will be performed in series. That is, the first calculation (in this case 1/2/1255) will be performed first. Once complete, the output coordinates of this calculation will be automatically submitted to the next method in the queue (in this case 1/2/6247) and so on until all requested methods have been performed.

**NOTE:** You can also queue up calculations to be performed with multiple programs, for example with Gaussian and Turbomole; the output from the previous calculation will automatically be converted to an appropriate input for the next program.

### Confirm Selection

Once the desired methods have been queued, select the 'Confirm' button to submit the chosen coordinate files to the chosen methods. Information will be shown as each coordinate file is prepared and then submitted. Once all files have been processed, the 'Successfully submitted x file(s)' line will appear:

Silico 1.0.0-pre.6	
Silico Calculation Submitter	
Program Output	
silico: INFO: Submitting file 'Benzene.com'...	
silico: INFO: Submitting file 'Naphthalene.xyz'...	
silico: INFO: Successfully submitted 2 file(s)	
< Confirm >	
< Back >> Confirm	
Arrow Keys: Navigate TAB: Next: SHIFT-TAB: Previous ENTER: Select ESC: Back	

## Silico 1.0.0 – Computational Chemistry Management – Quick Reference Guide

Submission is now complete, and Silico can be exited by pressing the 'ESC' button until the interface closes.

### Method Codes and Non-Interactive Submission

The codes that identify each method (such as 1/2/1255 and 1/2/6247 used in the above example) are constant, unique and will always refer to the same method. This property allows computations to be submitted quickly without using the interactive interface, if the codes are known in advance.

**NOTE:** Calculation codes are only constant for the same Silico version. Updates may add, remove or rearrange the available calculations, resulting in changes to the relevant codes. Users will be notified if this occurs.

To submit a calculation non-interactively, use the 'silico sub' command *without* the '-I' option, and instead make sure to use the '-c' (**c**ode) followed by the relevant code(s). For example, to recreate the submission shown in the interactive example above, the complete command would be:

```
> silico sub Benzene.com Naphthalene.xyz -c 1/2/1255 1/2/6247
silico: INFO: Parsing coordinate file 'Benzene.com'
silico: INFO: Parsing coordinate file 'Naphthalene.xyz'
silico: INFO: Submitting file 'Benzene.com'...
silico: INFO: Submitting file 'Naphthalene.xyz'...
silico: INFO: Successfully submitted 2 file(s)
```

### Custom Method Files

In addition to the methods built-in to Silico, calculations can also be performed according to a custom method file that can be written by the user (see Custom Methods for more information on writing method files). To submit to a method file, use the '-m' (**m**ethod) option to the 'silico sub' command:

```
> silico sub Benzene.com Naphthalene.xyz -m method.sim
silico: INFO: Parsing coordinate file 'Benzene.com'
silico: INFO: Parsing coordinate file 'Naphthalene.xyz'
silico: INFO: Submitting file 'Benzene.com'...
silico: INFO: Submitting file 'Naphthalene.xyz'...
silico: INFO: Successfully submitted 2 file(s)
```

Method files and codes can be freely intermixed and will be queued in series as normal:

```
> silico sub Benzene.com -c 1/2/1 -m method.sim -c 1/2/8
```

## Silico 1.0.0 – Computational Chemistry Management – Quick Reference Guide

### Common Codes

For reference, some of the more common method codes are printed here:

Code	Description	Comment
1/2/1	Performs the following at PBE1PBE/6-31G(d,p): <ul style="list-style-type: none"><li>• Ground optimisation (with frequency)</li><li>• TDA excited states (50/50)</li></ul>	Used for characterising organic fluorescent and TADF emitters
1/2/4	Performs the following at B3LYP/6-31+G(d,p), using SBKJC-VDZ-ECP for metals, in MeCN: <ul style="list-style-type: none"><li>• Ground optimisation (with frequency)</li><li>• TD-DFT excited states (50/50)</li><li>• Optimised triplet (with frequency)</li><li>• Single point singlet at the same geometry</li></ul>	Used for characterising organometallic phosphorescent emitters
1/3/5	Performs the following at SCS-CC2/cc-pVDZ: <ul style="list-style-type: none"><li>• Ground optimisation</li><li>• S<sub>1</sub> and S<sub>2</sub> excited states</li><li>• T<sub>1</sub> and T<sub>2</sub> excited states</li></ul>	Used for characterising multiple-resonant TADF emitters. See: Turbomole SCS-CC2 and MR-TADF
1/3/6	Performs the following at SCS-ADC(2)/cc-pVDZ: <ul style="list-style-type: none"><li>• Ground optimisation</li><li>• S<sub>1</sub> and S<sub>2</sub> excited states</li><li>• T<sub>1</sub> and T<sub>2</sub> excited states</li></ul>	Used for characterising multiple-resonant TADF emitters, faster than CC2. See: Turbomole SCS-CC2 and MR-TADF
1/2/2503	Optimisation (with frequency) at PBE1PBE/6-31G(d,p)	EZC recommended for organic emitters
1/2/2737	Optimisation (with frequency) at B3LYP/6-31G(d,p)	Popular method & basis set in the literature
1/2/2800	Optimisation (with frequency) at B3LYP/6-31+G(d,p), using SBKJC-VDZ-ECP for metals, in MeCN	EZC recommended for organometallic emitters
1/2/6247	TDA excited states at PBE1PBE/6-31G(d,p)	EZC recommended for organic emitters
1/2/12394	TD-DFT excited states at B3LYP/6-31+G(d,p), using SBKJC-VDZ-ECP for metals, in MeCN	EZC recommended for organometallic emitters
1/2/8587	TDA excited states (100 singlets) at PBE1PBE/6-31G(d,p)	Used for simulating absorption spectra of organic emitters
1/2/14734	TD-DFT excited states (100 singlets) at B3LYP/6-31+G(d,p), using SBKJC-VDZ-ECP for metals, in MeCN	Useful for simulating absorption spectra of organometallic emitters

## Turbomole SCS-CC2 and MR-TADF

Multi-resonant TADF emitters cannot be accurately modelled using conventional DFT. Instead, these emitters require a class of 'higher order' methods called coupled cluster (CC) methods (although see SCS-ADC(2)). Coupled cluster methods come in various flavours, some of which are implemented by both Gaussian and Turbomole, such as CCD, CCSD and CCSD(T), but the majority of these methods are extremely expensive (very slow). To overcome this, the approximate coupled-cluster singles and doubles model (CC2) uses a number of approximations to drastically increase the speed of computation, which makes it viable for (semi-) rapid characterisation. This CC2 method is only available in Turbomole, and so all MR-TADF type emitters must use Turbomole for the calculation of singlet and triplet energies.

Although faster than most CC methods, CC2 is still extremely slow compared to DFT. Therefore, it is recommended to first perform a DFT optimisation so the starting geometry can be brought closer to the final geometry, reducing the overall computation time. It is recommended that this DFT step be performed in Gaussian, so that calculated energies can be compared to other, non MR-TADF emitters.

To characterise an MR-TADF emitter, first queue up a full DFT auto characterisation with Gaussian 16 (code 1/2/1):

```

Silico 0.17.0-dev Calculation Browser

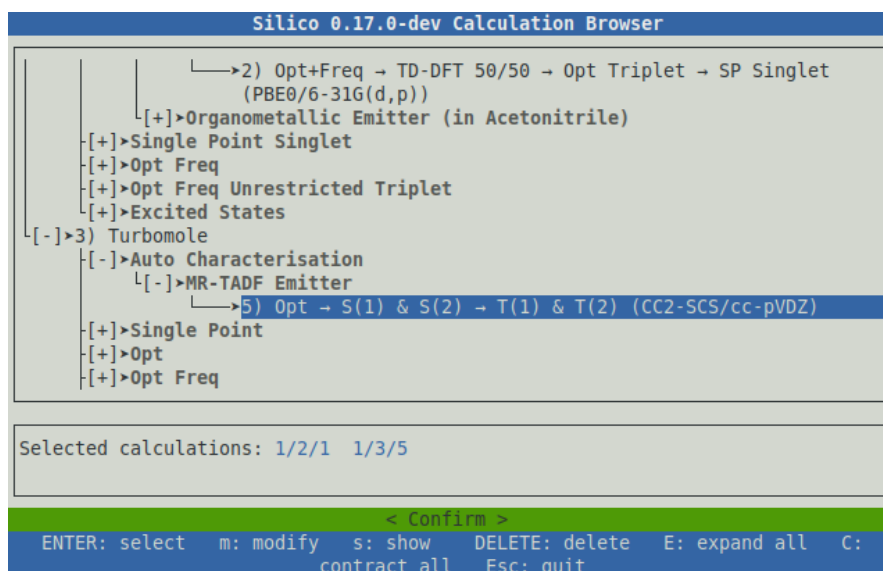
1) Singlenode SLURM (0 idle nodes, 71 (8%) idle CPUs)
  [+]>1) Gaussian 09
    [-]>2) Gaussian 16
      [-]>Auto Characterisation
        [-]>Organic Emitter
          1) Opt+Freq -> TDA 50/50 (PBE0/6-31G(d,p))
          2) Opt+Freq -> TD-DFT 50/50 -> Opt Triplet -> SP Singlet
             (PBE0/6-31G(d,p))
        [+]>Organometallic Emitter (in Acetonitrile)
      [+]>Single Point Singlet
      [+]>Opt Freq
      [+]>Opt Freq Unrestricted Triplet
      [+]>Excited States
    [+]>3) Turbomole

Selected calculations: 1/2/1

< Confirm >
ENTER: select  m: modify  s: show  DELETE: delete  E: expand all  C:
               contract all  Esc: quit
  
```

## Silico 1.0.0 – Computational Chemistry Management – Quick Reference Guide

Next, select Turbomole from the program list, and queue up a full auto characterisation using SCS-CC2 (code 1/3/5):



You can now submit the calculations as normal. This will perform a total of five calculations in series:

1. **Gaussian (DFT):** Optimisation of ground state and calculation of IR frequencies.
2. **Gaussian (DFT):** Calculation of the first 10 singlet and triplet excited states.
3. **Turbomole (CC2):** Optimisation of ground state.
4. **Turbomole (CC2):** Calculation of the first two singlet excited states.
5. **Turbomole (CC2):** Calculation of the first two triplet excited states.

**NOTE:** Unlike Gaussian, Turbomole cannot calculate both singlet and triplet excited states in the same calculation, so these are instead performed in series.

**NOTE:** Even on relatively small structures, full characterisation with CC2 can take several days or weeks to complete.

### SCS-ADC(2)

Another higher-order method supported by Turbomole is the second-order algebraic diagrammatic construction method, or ADC(2).<sup>1,2</sup> This method, although not a coupled cluster method like CC2, is similarly able to compute both single and double excitations, which is critical for the accurate prediction of excitation energies in MR-TADF like molecules. Compared to SCS-CC2, SCS-ADC(2) is on the order of 2.5x as fast while producing very similar results. As such, it is an attractive alternative, particularly for very large molecules. However, it should be noted that there is less support for ADC(2) in the current literature.

## Silico 1.0.0 – Computational Chemistry Management – Quick Reference Guide

To characterise an MR-TADF emitter (or other molecule of interest) with SCS-ADC(2), first queue up a full DFT auto characterisation with Gaussian 16 (code 1/2/1) as normal:

```
Silico 0.17.0-dev Calculation Browser

1) Singlenode SLURM (0 idle nodes, 71 (8%) idle CPUs)
  [+>1) Gaussian 09
  [->2) Gaussian 16
    [->Auto Characterisation
      [->Organic Emitter
        1) Opt+Freq → TDA 50/50 (PBE0/6-31G(d,p))
        2) Opt+Freq → TD-DFT 50/50 → Opt Triplet → SP Singlet
          (PBE0/6-31G(d,p))
      [+>Organometallic Emitter (in Acetonitrile)
    [+>Single Point Singlet
    [+>Opt Freq
    [+>Opt Freq Unrestricted Triplet
    [+>Excited States
  [+>3) Turbomole

Selected calculations: 1/2/1

< Confirm >
ENTER: select  m: modify  s: show  DELETE: delete  E: expand all  C:
              contract all  Esc: quit
```

Next, select SCS-ADC(2) (code 1/3/6) from the list of Turbomole calculations and confirm to submit:

```
Silico 0.19.6 Calculation Browser

1) Singlenode SLURM
  [+>1) Gaussian 09
  [+>2) Gaussian 16
  [->3) Turbomole
    [->Auto Characterisation
      [->MR-TADF Emitter
        5) Opt → S(1) & S(2) → T(1) & T(2) (CC2-SCS/cc-pVDZ)
        6) Opt → S(1) & S(2) → T(1) & T(2) (ADC(2)-SCS/cc-pVDZ)
      [+>Single Point
      [+>Opt
      [+>Opt Freq
      [+>Excited States
  2) BigMem SLURM

Selected calculations: 1/3/6

< Confirm >
ENTER: select  m: modify  s: show  DELETE: delete  E: expand all  C:
              contract all  Esc: quit
```

### Folder Structure

Silico will create a series of folders for each file and each calculation submitted. The topmost folder is named after the molecule submitted. Within this molecule directory, a separate folder is then created for each calculation, using the name of that calculation. If the same input file is submitted to multiple of the same calculation, a number will be appended to the calculation folder name to ensure each calculation is performed in a unique directory.

**NOTE:** The name of the 'molecule' directory is simply taken from the name of the input file; Silico does not try to automatically name structures.

Within each calculation directory, five sub folders are created. They are as follows:

#### Input

Contains the input file(s) for the calculation. The file(s) will be fully prepared for the calculation program, so they can be inspected to determine the specific parameters of the calculation. If an external basis set was requested (SBKJC-VDZ-ECP, for example) then this will be appended to the input file (if appropriate). If a non-native format was submitted (ChemDraw, for example), then this will have been converted appropriately.

**NOTE:** The file originally submitted is not stored in the Input directory; only the prepared input file is saved.

#### Output

Contains output files written by the calculation program.

**NOTE:** By default, Silico will automatically convert Gaussian .chk files to .fchk (before deleting the original .chk) in order to save file space. Occasionally, the original .chk file is required for post-analysis, in which case this option can be disabled.

#### Flags

Contains file flags, text files that convey information about the status of the calculation. See the section on **Monitoring Calculations/The Flags Folder** for more information about file flags.

#### Results

Contains text result files that are automatically written by Silico during post analysis.

#### Report

Contains a PDF report summary of the completed calculation, along with the rendered images used in the report. In addition, a smaller, mini report is also generated containing only atom coordinates, intended for easy inclusion into ESI.

# Monitoring Calculations

## The Flags Folder

Within each calculation directory, Silico manages a special folder with the name 'Flags'. This folder contains a number of empty text files where the name of each file conveys status about the calculation. These files are created and destroyed at key points in the calculation submission process, so the calculation can be monitored by observing which files are present (or absent) from the 'Flags' folder at any given moment.

To check the file flags, use the 'ls' command, followed by the path to the relevant Flags folder. For example:

```
> ls "Benzene/Gaussian 16/Opt Freq Gas Phase PBE1PBE 6-31G(d,p)/Flags"
RUNNING  STARTED
```

Here, two flags are present: 'RUNNING' and 'STARTED', which indicate that the calculation is currently ongoing.

Typing the full path to the flags folder is time consuming, so it is recommended to make use of autocompletion by using the 'tab' key. Begin typing the path, for example:

```
> ls Benzene/Gaussian\ 16/Opt
```

Then press the 'tab' key. Linux will automatically guess the rest of the filename (based on the available files that match what has already been typed):

```
> ls Benzene/Gaussian\ 16/Opt
> ls Benzene/Gaussian\ 16/Opt\ Freq\ Gas\ Phase\ PBE1PBE\ 6-31G\
(d\,p\)/
```

Then type 'Flags' to complete the path:

```
> ls Benzene/Gaussian\ 16/Opt
> ls Benzene/Gaussian\ 16/Opt\ Freq\ Gas\ Phase\ PBE1PBE\ 6-31G\
(d\,p\)/
> ls Benzene/Gaussian\ 16/Opt\ Freq\ Gas\ Phase\ PBE1PBE\ 6-31G\
(d\,p\)/Flags
```

**NOTE:** Autocompletion is a feature of Linux, not Silico, meaning you can use it with any Linux command.



## Silico 1.0.0 – Computational Chemistry Management – Quick Reference Guide

If the autocompletion function does not guess the filename, press 'tab' a second time to see a list of options:

```
> ls Benzene/Gaussian\ 16/Opt\ Freq\ Gas\ Phase\  
Opt Freq Gas Phase B3LYP 6-31G(d,p)/    Opt Freq Gas Phase PBE1PBE 6-  
31G(d,p) /  
> ls Benzene/Gaussian\ 16/Opt\ Freq\ Gas\ Phase\
```

Choose from the options listed by typing:

```
> ls Benzene/Gaussian\ 16/Opt\ Freq\ Gas\ Phase\  
Opt Freq Gas Phase B3LYP 6-31G(d,p)/    Opt Freq Gas Phase PBE1PBE 6-  
31G(d,p) /  
> ls Benzene/Gaussian\ 16/Opt\ Freq\ Gas\ Phase\ B
```

Then use 'tab' again to autocomplete the rest of the path:

```
> ls Benzene/Gaussian\ 16/Opt\ Freq\ Gas\ Phase\  
Opt Freq Gas Phase B3LYP 6-31G(d,p)/    Opt Freq Gas Phase PBE1PBE 6-  
31G(d,p) /  
> ls Benzene/Gaussian\ 16/Opt\ Freq\ Gas\ Phase\ B3LYP\ 6-31G(d,p) /
```

### List of Flags

The currently available file flags are as follows:

Name	Description
PENDING	The calculation has been submitted but has not yet begun; most likely because it is waiting in the queue.
STARTED	The calculation has begun. This flag is never deleted, so it is useful for confirming that the calculation at least started, even if it did not finish.
RUNNING	The calculation is currently ongoing.
SUCCESS	The calculation finished successfully.
CONVERGED	The optimisation converged successfully. This flag is only used for optimisation calculations.
NOT_CONVERGED	The optimisation did not converge successfully. This flag is only used for optimisation calculations.
CLEANUP	The calculation has finished (successfully or otherwise) and Silico is currently cleaning up (saving files etc).
ERROR	The calculation has stopped because an error occurred.
POST	The calculation has finished and Silico is currently performing post analysis (writing result and report files).
DONE	All work (including post-analysis) has been completed; Silico will not make any changes after this flag. It is safe to move, download or delete the calculation folder.

## Silico 1.0.0 – Computational Chemistry Management – Quick Reference Guide

### Multiple Calculations

Multiple calculations can be checked simultaneously by specifying multiple directories to the 'ls' command:

```
> ls "Benzene/Gaussian 16 Opt Freq Gas Phase PBE1PBE 6-31G(d,p)/Flags"
"Benzene/Gaussian 16 TDA 10 Singlets 10 Triplets Gas Phase PBE1PBE 6-
31G(d,p)/Flags"
Benzene/Gaussian 16 Opt Freq Gas Phase PBE1PBE 6-31G(d,p)/Flags:
CONVERGED DONE STARTED SUCCESS

Benzene/Gaussian 16 TDA 10 Singlets 10 Triplets Gas Phase PBE1PBE 6-
31G(d,p)/Flags:
DONE STARTED SUCCESS
```

This is more conveniently achieved using the Linux wildcard feature, using the asterisk character (\*). For example, to check all calculations in the Benzene directory:

```
> ls Benzene/*/*/Flags
Benzene/Gaussian 16/Opt Freq Gas Phase B3LYP 6-31G(d,p)/Flags:
CONVERGED DONE STARTED SUCCESS

Benzene/Gaussian 16 Opt Freq Gas Phase PBE1PBE 6-31G(d,p)/Flags:
CONVERGED DONE STARTED SUCCESS

Benzene/Gaussian 16 Single Point Singlet Gas Phase PBE1PBE
6-31G(d,p)/Flags:
DONE STARTED SUCCESS

Benzene/Gaussian 16 TDA 10 Singlets 10 Triplets Gas Phase PBE1PBE 6-
31G(d,p)/Flags:
DONE STARTED SUCCESS
```

Similarly, to check the status of all calculations of all molecules in the current directory, three wildcards can be used:

```
> ls */*/*/Flags
246tCzPPC/Gaussian 16 TDA Optimised S(1) B3LYP 6-31G(d,p)/Flags:
PENDING

26tCzPPC/Gaussian 16 TDA Optimised S(1) B3LYP 6-31G(d,p)/Flags:
RUNNING STARTED

2CzIPN/Gaussian 16 TDA Optimised S(1) B3LYP 6-31G(d,p)/Flags:
CONVERGED DONE STARTED SUCCESS

2CzPN/Gaussian 16 TDA Optimised S(1) B3LYP 6-31G(d,p)/Flags:
CONVERGED DONE STARTED SUCCESS

2CzTPN/Gaussian 16 TDA Optimised S(1) B3LYP 6-31G(d,p)/Flags:
CONVERGED DONE STARTED SUCCESS
...
```

## Post Analysis

### Results

Once a calculation is complete (the DONE flag has been set), result files will be available for analysis in the 'Results' folder. Perhaps the most important of these is the 'summary' file, which will be named something like 'Benzene.summary'. This file can be viewed on the server using the 'less' or 'cat' commands:

```
> less "Benzene/Gaussian 16 Opt Freq Gas Phase B3LYP 6-31G(d,p)/Results/Benzene.summary"
```

**NOTE:** When using the 'less' command, use the 'q' key to quit.

The summary file, as its name would suggest, contains a summary of the most important calculation results. An example file is displayed below:

```
> cat "Benzene/Gaussian 16 Opt Freq Gas Phase B3LYP
6-31G(d,p)/Results/Benzene.summary"
-----
|      Metadata      |
-----
Name: Benzene/Opt Freq Gas Phase B3LYP 6-31G(d,p)/Output/Benzene.log
Date: 30/07/2020 at 10:27:17
Duration: 0 days, 0 hours, 1 minutes
Computational package: Gaussian (2009+D.01)
Calculations: Optimisation, Frequencies
Methods: DFT
Functional: B3LYP
Basis set: 6-31G(d,p)
Multiplicity: Singlet
Charge: 0
Orbital spin: restricted
Success: True
Converged: True
Calculation temperature /K: 298.15
Calculation pressure /atm: 1.00

-----
|      Vibrational Frequencies      |
-----
No. vibrations: 30
No. negative frequency: 0

-----
|      Geometry      |
-----
Formula: C6H6
Exact mass /gmol-1: 78.0469
Molar mass /gmol-1: 78.1118
No. atoms: 12
Alignment method: Minimal
```

## Silico 1.0.0 – Computational Chemistry Management – Quick Reference Guide

```
X extension /Å: 5.00
Y extension /Å: 3.58
Z extension /Å: 2.85
Linearity ratio: 0.28
Planarity ratio: 0.20

-----
|      SCF Energy      |
-----
No. of steps: 31
Energy /eV: -6315.9237
Energy /kJmol-1: -609393.9974

-----
|      Orbitals      |
-----
HOMO eV: -5.73
LUMO eV: -1.88
HOMO/LUMO energy /eV: 3.85
No. virtual: 99
No. occupied: 21

-----
| Permanent Dipole Moment |
-----
Total /D: 1.23
Origin X /D: 0.00
Origin Y /D: 0.00
Origin Z /D: 0.00
Vector X /D: -0.25
Vector Y /D: 1.12
Vector Z /D: -0.43
X axis angle /°: 78.40
XY plane angle /°: 20.61
```

The remaining files in the 'Results' directory contain specific calculation results in CSV format. These too can be read in the same way as the summary file, but are more easily manipulated once downloaded to a personal computer.

**NOTE:** CSV format can be easily imported into a spreadsheet (such as Microsoft Excel, Libre Office or Google Sheets) and graphed, if desired.

### Available Result Files

The currently available result files are listed below (*FILE* is used as a stand-in for the actual filename):

Name	Description
<i>FILE</i> .summary	Human readable summary of all calculation results
<i>FILE</i> .atoms.csv	Optimised (if applicable) atom coordinates
<i>FILE</i> .orbitals.csv	Orbital numbers, labels and energies (restricted calculations only)
<i>FILE</i> .alpha.csv	Alpha orbital numbers, labels and energies (unrestricted calculations only)

## Silico 1.0.0 – Computational Chemistry Management – Quick Reference Guide

<i>FILE.beta.csv</i>	Beta orbital numbers, labels and energies (unrestricted calculations only)
<i>FILE.SCF.csv</i>	SCF (self consistent field) energy. For optimisations, includes the energy at each step which can be used to graph the calculation's convergence. The SCF energy is the HF (Hartree–Fock) or DFT energy.
<i>FILE.MP.csv</i>	Same as <i>FILE.SCF.csv</i> , but for Møller–Plesset energies. The energies are total (including HF and MP correction) and are from the highest MP level of the calculation.
<i>FILE.CC.csv</i>	Same as <i>FILE.SCF.csv</i> , but for coupled-cluster energies. The energies are total (including HF and CC correction, if applicable).
<i>FILE.ES.csv</i>	Excited state results (from TD-DFT, TDA etc). Includes energy, symmetry, multiplicity and orbital contributions.
<i>FILE.transitions.csv</i>	Excited state transitions. This information is also included in <i>FILE.ES.csv</i> , but the format here may be more convenient for directly comparing transitions.
<i>FILE.TDM.csv</i>	Excited state transition dipole moments.
<i>FILE.UV-Vis.csv</i>	A simulated UV-Vis spectrum in corrected units of wavelength (nm).
<i>FILE.absorptions.csv</i>	A simulated UV-Vis spectrum in uncorrected units of energy (eV).
<i>FILE.vibrations.csv</i>	Vibrational frequencies.
<i>FILE.IR.csv</i>	A simulated IR spectrum.
<i>FILE.SOC.csv</i>	Singlet/triplet spin-orbit coupling values.

### Report

Silico will also generate a PDF report file at the end of each calculation, this is available under the 'Report' folder. The report contains much the same data as is available in the result files, but in a more convenient and easily transferable format. In addition, the PDF report contains graphical results that can not be represented in text, including simulated spectra, convergence graphs, energy diagrams and orbital density images.

In addition to the report itself, all the rendered and generated image files are available under the 'Report/image' directory. For each 3D rendered structure image two images are available, a lower quality but smaller sized .jpg, and a higher quality, larger .png file.

**NOTE:** The PDF report itself uses the smaller .jpg files to ensure the file size is small enough to be sent by email. The higher quality .png files are included for publication purposes.

Finally, a smaller mini report is also available, named something like *FILE.atoms.pdf*, which only contains final atom coordinates. The intention of this mini-report is for it to be easily included in the ESI of journal papers.

## Spin-Orbit Coupling (SOC)

Silico can compute spin-orbit coupling values using a modified version of the PySOC<sup>3</sup> program. SOC will be computed automatically for time-dependant DFT calculations (with or without the Tamm Dancoff approximation) performed using Gaussian (other program are not yet supported). SOC will be calculated between all singlet states, including the ground, and all triplet states. In the pdf report, a summary of the SOC between S<sub>0</sub>/T<sub>1</sub> and S<sub>1</sub>/T<sub>1</sub> is available on the summary page:

### Spin-Orbit Coupling

$$\langle S_0 | H_{SO} | T_1 \rangle: 58.09 \text{ cm}^{-1}$$

$$\langle S_0 | \lambda | T_1 \rangle: 0.00$$

$$\langle S_1 | H_{SO} | T_1 \rangle: 0.00 \text{ cm}^{-1}$$

$$\langle S_1 | \lambda | T_1 \rangle: 0.00$$

Here, H<sub>SO</sub> is the SOC between the two given states, while λ is the first order mixing coefficient, given by:

$$\lambda = \frac{H_{SO}}{\Delta E_{ST}}$$

SOC values for all excited states are also available on the 'Table of Spin-Orbit Coupling' page later in the report:

*Table of Spin-Orbit Coupling*

Singlet	Triplet	SOC +1 /cm <sup>-1</sup>	SOC 0 /cm <sup>-1</sup>	SOC -1 /cm <sup>-1</sup>	SOC Root Sum Square /cm <sup>-1</sup>	H <sub>SO</sub> /eV	ΔE /eV	First Order Mixing Coefficient
S <sub>0</sub>	T <sub>1</sub>	41.0762	0.0010	41.0762	58.0905	0.0072	3.3687	0.0021
S <sub>0</sub>	T <sub>2</sub>	0.0000	0.0002	0.0000	0.0002	0.0000	5.9293	0.0000
S <sub>0</sub>	T <sub>3</sub>	40.3854	0.0012	40.3854	57.1136	0.0071	8.0721	0.0009
S <sub>0</sub>	T <sub>4</sub>	0.0002	11.3528	0.0002	11.3528	0.0014	8.3342	0.0002
S <sub>0</sub>	T <sub>5</sub>	9.8311	0.0002	9.8311	13.9033	0.0017	9.7027	0.0002
S <sub>0</sub>	T <sub>6</sub>	0.0000	0.0004	0.0000	0.0004	0.0000	10.2885	0.0000
S <sub>0</sub>	T <sub>7</sub>	0.0004	21.0802	0.0004	21.0802	0.0026	11.1793	0.0002
S <sub>0</sub>	T <sub>8</sub>	3.4940	0.0001	3.4940	4.9412	0.0006	12.2589	0.0000

Here, 'SOC +1', 'SOC 0' and 'SOC -1' are the spin-orbit coupling values for the degenerative triplet states with quantum numbers +1, 0 and -1 respectively.

## Analysis of Results

In addition to aiding calculation submission, Silico also includes a number of tools for analysing calculation result files. Analysis is performed using the 'silico res' (**result**) sub-program.

### Generating Summaries

Used without other arguments, Silico will generate a text summary of the given calculation result files. The content of this summary is the same as found in the *FILE.summary* file (see **Post Analysis/Results** section), and is useful for inspecting calculation files that were not submitted using Silico.

To analyse a file named 'Benzene.log', the command would be as follows:

```
> silico res Benzene.log
-----
|      Metadata      |
-----
Name: Benzene/TDA 10 Singlets 10 Triplets Gas Phase PBE1PBE 6-
31G(d,p)/Output/Benzene.log
Computational package: Gaussian (2009+D.01)
Calculations: Excited States"
Methods: DFT
Functional: PBE1PBE
...
```

To save the summary to a file, Linux file redirection can be used with the '>' character. For example, to save the summary to a file named 'Benzene.summary':

```
> silico res Benzene.log > Benzene.summary
```

**NOTE:** The standard redirection character (>) will overwrite the file if it already exists, deleting any existing data. To avoid this behaviour, use the double redirection character instead (>>).

### Analysis of Multiple Results

Although extracting and formatting results from a single calculation output file may be useful, the real strength of the result sub-program is being able to analyse and tabulate multiple calculation result files simultaneously.

This is achieved by specifying multiple files to 'silico result' with the '-a' (**table**) option. It is recommended that this command be paired with the 'less' command, which allows the considerable quantity of data to be scrolled, as follows:

```
> silico res Benzene.log "Methyl benzene.log" -a | less -S
```

## Silico 1.0.0 – Computational Chemistry Management – Quick Reference Guide

**NOTE:** Silico loads calculation result files in parallel, meaning there is little difference in time between reading one file or two (up to the number of cores in the computer).

The table format is useful for 'quickly' comparing the results from a handful of calculations, but may become cumbersome when comparing/analysing hundreds of results. For extended post analysis, the '-c' (**CSV**) option is recommended, which produces files that can be easily imported into a spreadsheet of choice.

To write tabulated CSV result files, use the '-c' option and redirect the output using the standard Linux redirection character '>'. For example, to write to a file named 'Results.csv':

```
> silico res Benzene.log "Methyl benzene.log" -c > Results.csv
```

When analysing a great many result files, typing each file name individually is cumbersome. Instead, using the Linux wildcard character (\*) is recommended. For example, to tabulate the results of all calculations in the current directory, the following command would be used:

```
> silico res */***/Output/*.log -c > Results.csv
```

The program may take some time to complete, depending on the number of calculations being analysed. The program will print the name of each file being read to keep track of progress:

```
> silico result */***/Output/*.log -c > Results.csv
silico: INFO: Parsing calculation result file 'Benzene/Gaussian 16 Opt
Freq Gas Phase B3LYP 6-31G(d,p)/Output/Benzene.log'
silico: INFO: Parsing calculation result file 'Benzene/Gaussian 16 Opt
Freq Gas Phase PBE1PBE 6-31G(d,p)/Output/Benzene.log'
silico: INFO: Parsing calculation result file 'Benzene/Gaussian 16
Single Point Singlet Gas Phase PBE1PBE 6-31G(d,p)/Output
/Benzene.log'
silico: INFO: Parsing calculation result file 'Benzene/Gaussian 16 TDA
10 Singlets 10 Triplets Gas Phase PBE1PBE 6-31G(d,p)/Output
/Benzene.log'
silico: INFO: Parsing calculation result file 'Benzene/Gaussian 16 TD-
DFT 10 Singlets 10 Triplets Gas Phase PBE1PBE 6-31G(d,p)/Output
/Benzene.log'
```

Once complete, the resulting file can be downloaded from the server and analysed at leisure.



# Generating Reports

## Normal Usage

The PDF reports automatically generated by Silico at the end of a calculation can also be generated manually by using the 'silico rep' (**report**) command. For Gaussian calculations, reports require both the main output file (.log) and also the checkpoint file (either .chk or .fchk), as image data is only stored in the latter. However, the report command will automatically search for checkpoint files in the same folder as the .log file, so long as they have the same name, so often only the .log file needs to be specified. No additional files need to be specified for Turbomole calculations. The report sub-program is used as follows:

```
> silico rep Benzene.log
```

The report will be written, by default, to a folder name 'Report' in the current folder. This can be changed using the '-O' (**output**) option:

```
> silico rep Benzene.log -O "Benzene Report"
```

Generating the report can take some time, especially for calculations involving excited states (TDA and TD-DFT). Silico will print status messages periodically to keep track of progress.

```
silico: INFO: Parsing calculation result '/Storage/oliver/Chemistry/St. Andrews PhD/Test Molecules/Benzene_triplet/Turbomole Opt Freq PBE0 (GD3BJ) 6-31G**/Output/Benzene_triplet.log'
silico: INFO: Generating report 'Report/Benzene_triplet.pdf'...
silico: INFO: Rendering Report/image/Benzene_triplet.SCF_graph.png to file(s)
silico: INFO: Rendering Report/image/Orbital Diagram/Benzene_triplet.alpha_HOMO_LUMO.png to file(s)
silico: INFO: Rendering Report/image/Orbital Diagram/Benzene_triplet.alpha_orbitals.png to file(s)
silico: INFO: Rendering Report/image/Orbital Diagram/Benzene_triplet.beta_HOMO_LUMO.png to file(s)
silico: INFO: Rendering Report/image/Orbital Diagram/Benzene_triplet.beta_orbitals.png to file(s)
silico: INFO: Rendering Report/image/Benzene_triplet.simulated_frequencies.png to file(s)
silico: INFO: Writing PDF file
silico: INFO: Done generating report 'Report/Benzene_triplet.pdf'
```

Once the report has been written, the line 'Done generating report' will be printed.

## Combined Reports

Reports can also be generated from more than one calculation result file, by specifying multiple log files:

```
> silico rep */Output/Benzene.log -O "Combi Report"
```

The results from each calculation will be parsed and then combined together to produce a single report. The order in which the calculation log files are given on the command line is important, as earlier calculations take precedence over later ones. If the same type of result, for example SCF

## Silico 1.0.0 – Computational Chemistry Management – Quick Reference Guide

energies, orbital energies or molecular vibrations etc is present in more than one of the calculation files, then later results will be ignored. Silico will generate a warning whenever results are skipped in this way:

```
> silico report */Output/Benzene.log -O "Combi Report"
...
silico: WARNING: UserWarning: Attempting to merge lists of atoms that
are not identical; non-equivalent atoms will be ignored
silico: WARNING: UserWarning: Attempting to merge lists of orbitals
that are not identical; non-equivalent orbitals will be ignored
silico: WARNING: UserWarning: Attempting to merge list of CC energies
that are not identical; non-equivalent energies will be ignored
silico: WARNING: UserWarning: Attempting to merge list of CC energies
that are not identical; non-equivalent energies will be ignored
silico: WARNING: UserWarning: Attempting to merge list of MP energies
that are not identical; non-equivalent energies will be ignored
silico: WARNING: UserWarning: Attempting to merge list of SCF energies
that are not identical; non-equivalent energies will be ignored
silico: WARNING: UserWarning: Attempting to merge list of SCF energies
that are not identical; non-equivalent energies will be ignored
...
```

As an exception, excited states are not skipped and are instead combined together. This allows, for example, both singlet and triplet excitations to be merged into a single report. This is particularly useful for Turbomole calculations, in which excited states of different multiplicities must be calculated separately.

### Automatic Combined Reports

When more than one calculation is performed in series, a combined report will be automatically generated once all calculations have been completed. Combined reports can be found in the 'Combined Reports' folder, within the molecule directory.

## Checking The Queue

The size of the server queue (the number of calculations waiting to be submitted) can be checked at any time using the 'silico sta' (silico **status**) command, which will display the following output:

```
> silico sta
```

Description	Status
[1] Single Node SLURM (destination)	0 (0%) idle nodes, 16 (1%) idle CPUs
[2] Debug SLURM (destination)	1 (100%) idle nodes, 32 (100%) idle CPUs
[3] Multi Node SLURM (destination)	0 (0%) idle nodes, 8 (0%) idle CPUs
[4] Local (series) (destination)	N/A (status not available)
[5] Local (parallel) (destination)	N/A (status not available)

This table lists one row for each available queue. Typically the top queue, Singlenode SLURM, is of most interest as this is where the majority of calculations are submitted to. The rightmost column describes the current size of the queue: the number of 'idle nodes' indicates the number of machines in the cluster that are currently 'empty' (doing nothing), while the number of 'idle CPUs' indicates the number of CPUs that are freely available. The smaller the number of idle CPUs, the busier the queue and the longer a calculation is likely to have to wait before starting.

**NOTE:** Calculations can always be submitted regardless of the queue size; if insufficient CPUs are currently available then new calculations will be added to the back of the queue and will be started automatically once sufficient resources become available.

## Referencing Silico

Silico is currently unpublished software and so you do not need to reference Silico directly in your work. However, Silico depends on a number of third party libraries and programs in order to function and it is important that you cite these appropriately.

The references you are required to cite will depend on which features of Silico you have used:

### Calculation submission

Interconversion of file formats is performed by the Open Babel library. All work that uses Silico to submit calculations should cite both Open Babel<sup>4</sup> and Pybel.<sup>5</sup>

### Calculation results

Extraction and processing of result files is performed with the aid of the cclib library. All work that uses calculation results processed by Silico should cite cclib.<sup>6</sup> This includes results taken from both calculation reports and result text files.

### Molecular and orbital images

Image rendering is performed by the VMD program and the Tachyon ray tracer. All work that directly includes 3D, rendered images of molecules, orbitals or spin densities in the body of the publication (or ESI) should cite both VMD<sup>7</sup> and Tachyon.<sup>8</sup>

### Graphs

Graphs are drawn using the Matplotlib library. All work that directly includes graphs produced by Silico in the body of the publication (or ESI), including the 1D excited states energy graph, should cite Matplotlib.<sup>9</sup>

### Spin-orbit coupling

Spin-orbit coupling (SOC) is calculated using a modified version of the PySOC program. All work that includes SOC values calculated using Silico should cite the original PySOC publication.<sup>3</sup>

## Example Reference

An example reference suitable for inclusion in the methodology section of a publication is included below. Irrelevant sections should be deleted as appropriate:

Calculations were submitted and processed using in-house developed software which incorporates a number of publicly available software libraries, including: cclib<sup>6</sup> for parsing of result files, VMD<sup>7</sup>/Tachyon<sup>8</sup> for 3D rendering, Matplotlib<sup>9</sup> for drawing of graphs, Open Babel<sup>4</sup>/Pybel<sup>5</sup> for file interconversion and PySOC<sup>3</sup> for the calculation of spin-orbit coupling.

## Bibliography

- 1 A. B. Trofimov and J. Schirmer, *J. Phys. B At. Mol. Opt. Phys.*, 1995, **28**, 2299–2324.
- 2 J. Schirmer, *Phys. Rev. A*, 1982, **26**, 2395–2416.
- 3 X. Gao, S. Bai, D. Fazzi, T. Niehaus, M. Barbatti and W. Thiel, *J. Chem. Theory Comput.*, 2017, **13**, 515–524.
- 4 N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch and G. R. Hutchison, *J. Cheminform.*, 2011, **3**, 33.
- 5 N. M. O'Boyle, C. Morley and G. R. Hutchison, *Chem. Cent. J.*, 2008, **2**, 5.
- 6 N. M. O'Boyle, A. L. Tenderholt and K. M. Langner, *J. Comput. Chem.*, 2008, **29**, 839–845.
- 7 W. Humphrey, A. Dalke and K. Schulten, *J. Mol. Graph.*, 1996, **14**, 33–38.
- 8 J. Stone, Computer Science Department, University of Missouri-Rolla, 1998.
- 9 J. D. Hunter, *Comput. Sci. Eng.*, 2007, **9**, 90–95.