
job-script.py

```
1  # This is the script for one job instance, parametrized by $SGE_TASK_ID.
2  import bootstrap
3  import matplotlib.pyplot as plt
4  import time
5  import datetime
6  import numpy as np
7  from matplotlib.backends.backend_pdf import PdfPages
8  from revised_ising_gap import *
9  import sys
10
11 # Read the arguments given and assign them their correct variables names.
12 # First argument will be $SGE_TASK_ID.
13 SGE_TASK_ID = sys.argv[1]
14 k_start = sys.argv[2]
15 k_stop = sys.argv[3]
16 l_start = sys.argv[4]
17 l_stop = sys.argv[5]
18 m_start = sys.argv[6]
19 m_stop = sys.argv[7]
20 n_start = sys.argv[8]
21 n_stop = sys.argv[9]
22
23 # Convert these raw numbers into Python lists.
24 if k_stop - k_start == 0:
25     k_range = [k_start]
26 else:
27     k_range = np.arange(k_start, k_stop + 1, 1).tolist()
28 if l_stop - l_start == 0:
29     l_range = [l_start]
30 else:
31     l_range = np.arange(l_start, l_stop + 1, 1).tolist()
32 if m_stop - m_start == 0:
33     m_range = [m_start]
34 else:
35     m_range = np.arange(m_start, m_stop + 1, 1).tolist()
36 if n_stop - n_start == 0:
37     n_range = [n_start]
38 else:
39     n_range = np.arange(n_start, n_stop + 1, 1).tolist()
40
41 # Instantiate an ising_gap object, which has default dim, gap, sig_values, eps_values & 'cutoff'.
42 ising_gap = Ising_Gap()
43
44 # Generate all specified grids, store them in the class 'table' attribute.
45 ising_gap.iterate_parameters(k_range, l_range, m_range, n_range)
46
47 # Save the contents of 'table' to an executable python file,
48 ising_gap.save_to_file(SGE_TASK_ID.__str__())
```