

```

import time
import numpy as np
from joblib import Parallel, delayed
from bdm import BDM
from bdm.utils import slice_dataset

bdm = BDM (ndim = 1)

kernels = 4 #number of kernels
long_max = 5000 #number of sequences to consider
time_lib = np.zeros ((long_max - 11)) #time for BDM
time_mi_lib = np.zeros ((long_max - 11)) #time for BDMNN
lib = np.zeros ((long_max - 11)) #complexity by BDM
mi_lib = np.zeros ((long_max - 11)) #complexity by BDMNN
longs = np.zeros ((long_max - 11))

for i in range (12, long_max + 1) :
    prueba = np.random.choice (2, kernels*i).reshape (1, kernels*i)
    start = time.process_time ()
    mi_lib[i - 12] = kolmo (prueba)
    time_mi_lib[i - 12] = time.process_time () - start

    if len (prueba[0]) >= 48 :
        counters_demo = Parallel (n_jobs = 4) (delayed (bdm.count_and _lookup) (d) for d in
slice_dataset (prueba[0], (round (len (prueba[0])/kernels),)))
        start = time.process_time ()
        lib[i - 12] = bdm.compute_bdm(*counters_demo) time_lib[i-12]=time.process_time()-
start
    else:
        start=time.process_time() lib[i-12]=bdm.bdm(prueba[0])
        time_lib[i-12]=time.process_time()-start longs[i-12]=kernels*i

```