

# Geometric-Methods-in-Data-Analysis

---

## Circumventing the Distance Concentration Phenomena

Flavie Vampouille – Paul Vercoustre

Description. Several strategies can be developed to deal with distance concentration phenomena. One of them, seen in class, consists in using suitable norms, and to project to lower dimensional spaces. Another one, proposed in CTP11

(<https://github.com/paulvercoustre/Geometric-Methods-in-Data-Analysis/blob/master/References.md>) , consists in using a biasing potential which aims at focusing on the most informative distances only. In this project, we aim at applying the procedure from CTP11


(<https://github.com/paulvercoustre/Geometric-Methods-in-Data-Analysis/blob/master/References.md>) to a different molecular data set, namely an ensemble of conformations of a protein model known as BLN69 RDRC16 (<https://github.com/paulvercoustre/Geometric-Methods-in-Data-Analysis/blob/master/References.md>) . In a nutshell, BLN69 is a linear chain of 69 beads; since each bead has 3 cartesian coordinates, a conformation is defined by a point in dimension  $d = 3 \times 69 = 207$ . To each conformation, one can also associate an energy, which will be given along with the conformations. Finally, to measure the distance between two conformations, we shall use the least root mean square deviation [http://sbl.inria.fr/doc/Molecular\\_distances-user-manual](http://sbl.inria.fr/doc/Molecular_distances-user-manual) ([http://sbl.inria.fr/doc/Molecular\\_distances-user-manual](http://sbl.inria.fr/doc/Molecular_distances-user-manual)) . [html](http://sbl.inria.fr/doc/Molecular_distances-user-manual).

N.B on workflow: For this project we chose to use the pre-compiled static SBL programs provided at <http://sbl.inria.fr/applications> (<http://sbl.inria.fr/applications>) rather than compiling the entire SBL library which happens to be quite challenging. Therefore we have run several different packages on a VM CentOS which we combined with some python code to answer the given tasks.

### Question 1

**An ensemble of  $N 10^6$  local minima of the BLN69 protein model can be found at <http://sbl.inria.fr/data-models>. This set is denoted  $S$  in the sequel. To get familiar with this data set, select a reasonable number of local minima with low energy, and display them in 2D using multi-dimensional scaling (MDS). For example, you may focus on the 10 lowest local minima. This set is denoted  $T$  in the sequel.**

In order to generate set  $T$ , we import set  $S$  and choose the 10 conformations with the lowest associated energies:

```
 # find the index of the 10 lowest local minima
idx = np.argpartition(E_S, 10)

# T is the corresponding matrix of coordinates
T = S[idx[0:10],]
```

We obtain this ([https://github.com/paulvercoustre/Geometric-Methods-in-Data-Analysis/blob/master/data/10\\_local\\_minima.txt](https://github.com/paulvercoustre/Geometric-Methods-in-Data-Analysis/blob/master/data/10_local_minima.txt)) set T of conformations

We run the SBL – Conformational Analysis package with set T using the following command line:

```
sbl-conf-ensemble-analysis-lrmsd.exe --points-file /home/cloudera/Shared/10_local_minima.txt --pairwise-distances
```

Using the output text file obtained we run a multi-dimensional scaling in sklearn:

```
from sklearn import manifold
from sklearn.metrics import euclidean_distances
from adjustText import adjust_text

seed = 1 #generate reproducible results

mds_lrmsd = manifold.MDS(n_components=2, max_iter=3000, eps=1e-9, random_state=seed,
                        dissimilarity="precomputed", n_jobs = -1, verbose = True)

T_dist_mds = mds_lrmsd.fit_transform(T_dist)

fig, ax = plt.subplots()
ax.scatter(T_dist_mds[:, 0], T_dist_mds[:, 1])

texts = []
for i, txt in enumerate(idx[0:10]):
    texts.append(ax.text(T_dist_mds[:, 0][i], T_dist_mds[:, 1][i], txt))
adjust_text(texts)

plt.xticks([-1,-0.5,0,0.5,1])
plt.yticks([-1,-0.5,0,0.5,1])
plt.title("2D MDS on 10 lowest local minima of BNL69")
#plt.show()
plt.savefig('MDS_Q1.png')
```

Here we used a seed in order to be able to compare our results later in the project once we have applied the sketch-map method.

You can find the full code relative to this question here ([https://github.com/paulvercoustre/Geometric-Methods-in-Data-Analysis/blob/master/code/Task1\\_Notebook.ipynb](https://github.com/paulvercoustre/Geometric-Methods-in-Data-Analysis/blob/master/code/Task1_Notebook.ipynb))

We obtain the following plot:



## Question 2

**We wish to analyze pairwise distances between selected conformations. Since N precludes using all pairs, propose two procedures to:**

**Select a subset S1 of n conformations by retaining the low energy conformations only. Hint: you may use topological persistence, see e.g. CDM+15.**

In order to generate S1 we used the "Cluster Analysis" package from the SBL library, more specifically we applied a Morse theory based strategy with the program 'sbl-cluster-MTB-euclid.exe'. We call the following command:

```
➤ ./sbl-cluster-MTB-euclid.exe --points-file /home/cloudera/GMDA/hybrid-TRRT-BH-BLN__minima.txt --num-neighbors=20 --persistence-threshold=.1 --verbose
```

The algorithm ran for ~ 15 mins and produced these (<https://github.com/paulvercoustre/Geometric-Methods-in-Data-Analysis/tree/master/data/S1>) files with 1103 points.

```
➤ General Statistics:
```

```
Times elapsed for computations (in seconds):  
-- Cluster Engine: 938.920263  
Total: 938.920263
```

**Select a subset S2 of n conformations maximizing the distances between the conformations selected. Hint: you may use the smart seeding procedure used in k-means.**

To create subset S2, we again use the "Cluster Analysis" package from SBL library, this time applying k-means with a non random/smart seeding procedure and k = 1103 in order to have S1 & S2 of equal size.

We use the following command line:

```
➤ ./sbl-cluster-k-means-euclid.exe --k-means-k 1103 --points-file /home/cloudera/GMDA/hybrid-TRRT-BH-BLN__minima.txt --k-means-selector=plusplus --verbose
```

The algorithm ran for ~ 3h 15min and produced these (<https://github.com/paulvercoustre/Geometric-Methods-in-Data-Analysis/tree/master/data/S2>) files.

```
➤ General Statistics:
```

```
Times elapsed for computations (in seconds):  
-- Cluster Engine: 11812.583215  
Total: 11812.583215
```

The resulting points are the centroids of the 1103 clusters. In order to find the Fermat-Weber points (i.e. actual conformations), we need to find the nearest neighbours of the centroids within their respective cluster. To do so, we use the following code:

```
➤ from scipy.spatial import distance  
  
# distance between protein and the center of the affected cluster  
dist = np.zeros_like(S2clusters)  
for i in xrange(len(S2clusters)):  
    dist[i] = distance.euclidean(S[i], S2centers[S2clusters[i]])
```

```

# choose the protein the more close to the center to maximize distance in S2
S2 = np.zeros_like(S2centers)
first_visit = np.zeros(S2centers.shape[0])

min_dist = np.zeros(S2centers.shape[0])
for k in xrange(S2centers.shape[0]):
    for j in xrange(S.shape[0]):
        if S2clusters[j] == k:
            if first_visit[k] == 0:
                S2[k] = S[j]
                min_dist[k] = dist[j]
                first_visit[k] = 1
            else:
                if dist[j] < dist[k]:
                    S2[k] = S[j]

```

You can find the full code relative to this question here ([https://github.com/paulvercoustre/Geometric-Methods-in-Data-Analysis/blob/master/code/Task2\\_Notebook.ipynb](https://github.com/paulvercoustre/Geometric-Methods-in-Data-Analysis/blob/master/code/Task2_Notebook.ipynb))

### Question 3

**Using functionalities from the Molecular distances package from the SBL ([http://sbl.inria.fr/doc/Molecular\\_distances-user-manual.html](http://sbl.inria.fr/doc/Molecular_distances-user-manual.html)), produce a plot identical to CTP11, Fig 1 (C) for the sets S1 and S2.**

To get a first idea of the distributions of distances between pairs of frames in S1 and S2, we calculate the matrix of distances with `sbl-conf-ensemble-analysis.exe` program of the SBL library:

```

$ ./sbl-conf-ensemble-analysis-lrmsd.exe --points-file /home/cloudera/Shared/S1/S1.
txt --pairwise-distances

```

We obtain `S1_dist_matrix` and `S2_dist_matrix`, the 1103 x 1103 matrices of pairwise distances according the LRMSD. We then display the conformations from both sets in 2D using multi-dimensional scaling as in question 1.

You can find the code relative to this step here ()

These plots clearly show a big concentration with some spare points around.



We then want to study the distribution of distances between pairs of frames in a more accurate way. To achieved that goal we plot the histograms of pairwise distances between pairs of frames for both S1 & S2.

First we calculate the pairwise distance using the "Molecular Distances" package from the SBL library. More specifically we use `sbl-lrmsd-all-pairs.exe` program with:

```

$ for d in {1..2}; do ./sbl-lrmsd-all-pairs.exe --points-file /home/cloudera/Desk

```

```
op/GMDA/DATA/S${d}.txt --all-distances; mv all_distances.txt S${d}_dist.txt; done
```

We obtain S1\_pairwise\_dist & S2\_pairwise\_dist the 607740 x 3 matrices of pairwise distances where each line represents a combination of frames and the distance associated.

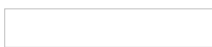
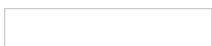
We then plot the histograms of distances between pairs of frames:

```
# log histogram
pylab.xlim([0,3.5])
h_log = plt.hist(S1_dist[:,2], bins=200, color='blue', histtype='stepfilled', log=
'False')
plt.yscale('log')
plt.title("Log-distribution of distances between pairs of frames - S1")
plt.xlabel('Distances')
plt.ylabel('Frequences of distances')
#plt.show()
plt.savefig('S1_all_logdist.png')

# log-histogram curve
pylab.xlim([0,3.5])
sns.distplot(S1_dist[:,2], hist=False)
plt.yscale('log')
plt.title("Histogram curve for the log-distribution of distances between pairs of f
rames - S1")
plt.xlabel('Distances')
plt.ylabel('Frequences of distances')
#plt.show()
plt.savefig('S1_all_logdist_curve.png')
```

You can find the full code relative to this task here ([https://github.com/paulvercoustre/Geometric-Methods-in-Data-Analysis/blob/master/code/Task3\\_Notebook.ipynb](https://github.com/paulvercoustre/Geometric-Methods-in-Data-Analysis/blob/master/code/Task3_Notebook.ipynb)) .

We obtain these plots with a log y scaling



**Question 4 Analyse the distributions of pairwise distances for the sets S1 and S2. You may proceed in 2 directions:**

**As in CTP11, check whether portions of the distribution correspond to distances between random points drawn according to a Gaussian distribution.**

Understanding proximities and distances between data lying in high dimensional space is often not intuitive and difficult to apprehend. Therefore we cannot expect to understand the BLN69 conformations by projecting them in a 2D dimension and plotting them (like we did as an introduction in task 3 to catch a first glimpse of the data). A good way to know if it is possible to represent the data in a lower dimensionally space is to plot the histogram of distances between pairs of frames (task 3)

To study the repartition of data in the plots obtained in task 3 we generate different objects:

1.1 We create a matrix of 1103 individuals following an isotropic multivariate gaussian distribution of mean zero. For S1 we use a variance of 0.1 and for S2 we set the variance to 0.2

```
# generate 1103 individuals following an isotropic multivariate gaussian with d = 207
mean = np.zeros(207)

cov = 0.1*np.eye(207)
x = np.random.multivariate_normal(mean, cov,1103)
x = np.insert(x, [0], 207, axis = 1)
np.savetxt("gaussianS1.txt",x,delimiter=" ",fmt='%.5f')

cov = 0.2*np.eye(207)
x = np.random.multivariate_normal(mean, cov,1103)
x = np.insert(x, [0], 207, axis = 1)
np.savetxt("gaussianS2.txt",x,delimiter=" ",fmt='%.5f')
```

1.2 We compute the pairwise distances on this data (after adding a first column that indicate dimensionality) with the sbl-lrmsd-all-pairs.exe program.

```
for d in {1..2}; do ./sbl-lrmsd-all-pairs.exe --points-file /home/cloudera/Desktop/GMDA/DATA/gaussianS${d}.txt --all-distances; mv all_distances.txt gaussianS${d}_all_distances.txt; done
```

1.3 And we add the histogram of pairwise distances obtained on the previous plots.

```
sns.set_style("whitegrid")
pylab.xlim([-0.2,3.5])
sns.distplot(S1_all_dist[:,2], hist=False, color="red")
sns.distplot(gaussianS1_all_dist[:,2], hist=False, color="black")
sns.distplot(uniformS1_all_dist[:,2], hist=False, color="grey")
plt.yscale('log')
plt.title("Kernel density estimate of the distribution of S1 distances between pairs of frames")
plt.xlabel('Distances')
plt.ylabel('Frequencies of distances')
plt.show()
```

2.1 We create a matrix of 1103 individuals and 207 features with a uniform distribution. To choose the range of the distribution we take a look at the data in S1 and S2 and observe that most of the coordinates are between -3 and 3. So we try several variations around these values and retain for S1 a range of -2.3, 2.3 and for S2 a range of -2.5 and 2.5.

```
x = np.random.uniform(-2.3,2.3,(1103,207))
x = np.insert(x, [0], 207, axis = 1)
np.savetxt("uniformS1.txt",x,delimiter=" ",fmt='%.5f')

x = np.random.uniform(-2.4,2.4,(1103,207))
```

```
x = np.insert(x, [0], 207, axis = 1)
np.savetxt("uniformS2.txt", x, delimiter=" ", fmt= '%.5f')
```

2.2 We then run the sbl-lrmsd-all-pairs.exe program on this data (after adding a first column that indicate dimensionality).

```
```for d in {1..2}; do ./sbl-lrmsd-all-pairs.exe --points-file
/home/cloudera/Desktop/GMDA/DATA/uniformS${d}.txt --all-distances; mv
all_distances.txt uniformS${d}_all_distances.txt; done
```

2.3 And we add the histogram of pairwise distances obtained on the previous plots

```
```python
sns.set_style("whitegrid")
pylab.xlim([-0.1,4])
sns.distplot(S2_all_dist[:,2], hist=False, color="red")
sns.distplot(gaussianS2_all_dist[:,2], hist=False, color="black")
sns.distplot(uniformS2_all_dist[:,2], hist=False, color="grey")
plt.yscale('log')
plt.title("Kernel density estimate of the distribution of S1 distances between pairs of frames")
plt.xlabel('Distances')
plt.ylabel('Frequencies of distances')
plt.show()
```

We can see that for small distances and long range distances, the distribution is characteristic of gaussian and uniform distributed points in the 207-dimensional space. Hence there is only a part of the distances between pairs of frames that holds real information:

- for values of distances around 1 (more or less depending on S1 or S2 set) the repartition of distances resemble that of a isotropic Gaussian with a standard deviation of 0.5 distribution in the 207-dimensional space;
- for the long range distances, it looks more like distances obtained from a uniform distribution of points in a 207-dimensional space.



We have shown that certain features of the distribution of distances are characteristics of randomly or uniformly distributed points in the full dimensional space. Therefore, not all distances are informative and we deduce from our plots that the interesting distances to be studied are those:

- between 1.5 and 3 for S1
- between 1.7 and 2.5 for S2

We can find the full code relative to this task here ([https://github.com/paulvercoustre/Geometric-Methods-in-Data-Analysis/blob/master/code/Task4\\_Notebook.ipynb](https://github.com/paulvercoustre/Geometric-Methods-in-Data-Analysis/blob/master/code/Task4_Notebook.ipynb))