

# Foundations of Geometric Methods in Data Analysis: Projects

Frederic.Cazals@inria.fr, Steve.Oudot@inria.fr

Academic year: 2016-2017

## Contents

<b>0</b>	<b>Projects: general recommendations</b>	<b>2</b>
<b>1</b>	<b>Search algorithms in metric trees</b>	<b>4</b>
<b>2</b>	<b>kd-trees and adaptation to the intrinsic dimension</b>	<b>5</b>
<b>3</b>	<b>The Earth Mover Distance and linear programming</b>	<b>6</b>
<b>4</b>	<b>Circumventing distance concentration phenomena</b>	<b>7</b>
<b>5</b>	<b>Multivariate two-sample tests</b>	<b>8</b>
<b>6</b>	<b>Detection of differences via feedback analysis</b>	<b>9</b>
<b>7</b>	<b>Mode-seeking for detecting metastable states in protein conformations</b>	<b>10</b>
<b>8</b>	<b>Analysis of the NBA postseason statistics using topological methods</b>	<b>11</b>

Procedure to select the projects:

- Groups of two students must register on the following doodle <http://doodle.com/poll/g6ymqt7c6a4gimuf>  
NB: Every option can be chosen by maximum 3 participant(s).
- When choosing a project, make sure to list to two lastnames.

Deadline to return your work: February the 26th.

See procedure in Section 0.

## 0 Projects: general recommendations

**Returning your work.** You will upload your production (report, source code etc) using the Inria gforge at <https://gforge.inria.fr/>.

The project fgmda-2016-17 has been created, and it contains the following directories, which match the pairs found in the doodle:

Brier\_Constantinescu  
Khayat\_Dubois  
WU\_Gopalakrisna  
AbouAmal\_Amar  
Chaton\_Thuleau  
Gette\_Xiang  
Roburin\_Dirhoussi  
Rong\_Guo\_Li  
Vampouille\_Vercoustre  
SAMOU\_PUPIN  
Seksoui\_Salaun  
Woo\_Ouazzani  
Ly\_Reille  
Sveen\_Petre  
Costet\_Fouret  
XU\_WANG  
Bouvier\_Jamin\_Changeart  
Jardillier\_Libeer  
Jacomet\_Moret  
JIN\_GUO\_ZHANG  
Leblanc\_Morand  
AmineJaiHokimi\_BatisteHaller

To interact with the gforge, each student will have to:

- register i.e. create a personal account
- send an inquiry to F. Cazals or S. Oudot to be invited to the projects hosting all productions
- generate a pair of public/private ssh keys. then, once logged into one's gforge account: click on 'My page' > Account maintenance and put your ssh public key in there.
- finally, to retrieve the working directory – illustration on the first binom:

```
svn checkout svn+ssh://MYLOGIN@scm.gforge.inria.fr/svnroot/fgmda-2016-17/Brier_Constantinescu
```

or

```
svn checkout --username MYLOGIN  
https://scm.gforge.inria.fr/authscm/fcazals/svn/fgmda-2016-17/Brier\_Constantinescu
```

**Projects with coding: instructions.** Several projects require coding in C++ and / or python. The following recommendations are in order:

- Program options. Programs should have command-line options properly documents, in order for users to easily pass different arguments. In python, one can use the package OptionParser, see <https://docs.python.org/2/library/optparse.html>. In C++, boost program options are highly recommended, see [http://www.boost.org/doc/libs/1\\_62\\_0/doc/html/program\\_options.html](http://www.boost.org/doc/libs/1_62_0/doc/html/program_options.html).

- Output of executions. Ad hoc output are not easily dealt with, unless one knows how to parse the output. Albeit verbose, xml files have two major advantages: (i) the tags allow one to comment on the output, and (ii) xml files are easily parsed with XML query language.

For C++ users, boost provides serialization mechanisms making it very easy to dump XML files. For a starting point, check out [http://www.boost.org/doc/libs/1\\_62\\_0/libs/serialization/doc/index.html](http://www.boost.org/doc/libs/1_62_0/libs/serialization/doc/index.html).

For python users, dictionaries are also easily serialized. See e.g. <https://docs.python.org/2/library/json.html>.

- Compilation for C++ code. Provide a CMakeLists.txt, from which the instructors will easily compile.
- Experiments. If you run several experiments, for example by varying one (or several) parameter(s), as requested in several projects, it is highly recommended to use a *batch manager* (BM). From a simple text file listing the options and their values, a batch manager handles all executions, by passing the relevant options on the command line.

You can for example use the BL from the Structural Bioinformatics Library, see [http://sbl.inria.fr/doc/Batch\\_manager-user-manual.html](http://sbl.inria.fr/doc/Batch_manager-user-manual.html).

In passing, if you have serialized your data structures, you can easily compute statistics using PALSE, see <http://sbl.inria.fr/doc/PALSE-user-manual.html>.

# 1 Search algorithms in metric trees

**Description.** The goal of this project is to carry out experiments on the performances of search procedures for metric trees [1].

**Tasks.** In the following, programming in C++ is highly recommended.

1. Provide an implementation of metric trees, with two search strategies: the exact one which uses a pruning condition to limit the number of nodes visited, and the defeatist style strategy.
2. Generate random points in a fixed dimensional Euclidean space, and use these to build a metric tree storing them. Then, run random queries on the tree built. Compare the number of nodes visited for the two search strategies. Run various experiments by:
  - varying the dimension of the ambient space – using fully dimensional data eg data drawn according to a mixture of gaussians.
  - varying the dimension of the ambient space while keeping the intrinsic dimension data constant.
3. Repeat the previous experiment for one type of complex data of your choice. Two possible options are: images to be compared with the earth mover distance; molecular conformations to be compared with the so-called lest RMSD. For molecular conformations, on can ease the energy landscape explorer from the Structural Bioinformatics Library, see [http://sbl.inria.fr/doc/Landscape\\_explorer-user-manual.html](http://sbl.inria.fr/doc/Landscape_explorer-user-manual.html)
4. In designing a metric tree, the choice of the pivot choice is critical. In class, we discussed the randomized choice. Provide a deterministic strategy aiming at minimizing the number of nodes visited during a search operation.

**Contact.** Frederic Cazals: [frederic.cazals@inria.fr](mailto:frederic.cazals@inria.fr)

## 2 kd-trees and adaptation to the intrinsic dimension

**Description.** We have seen in class that random projection trees adapt to the intrinsic dimension of data, with in particular properties on the diameter of point set stored in nodes of the tree. We have also seen that kd-trees can fail to reduce the diameter, or may require a large number of iterations to do so. Such pathological examples, though, use specific constructions using coordinate axis. Following [2], the goal of this project is to explore a simple idea: can kd-tree adapt to the intrinsic dimension of one randomly rotates the data stored?

**Tasks.** In the following, programming in C++ is highly recommended.

1. Develop a procedure to compute the diameter of a set of  $n$  points in  $\mathbb{R}^d$ . (NB: the diameter is the maximum distance between two points.) What is its complexity? Is it optimal?
2. Develop a procedure performing a random rotation of a point set in  $\mathbb{R}^d$ . NB: we have seen in class how to generate a random orthonormal matrix.
3. Implement a kd-tree data structure, with a jittered split – one splits at a random point in an interval around the median. See details in [2].
4. Run tests by:
  - varying the intrinsic dimension of the data,
  - varying the width of the interval used for the jittered split

In running these experiments, you will provide evidence on Thm2 from [2], and stress the role of the jittered split

**Contact.** Frederic Cazals: frederic.cazals@inria.fr

### 3 The Earth Mover Distance and linear programming

**Description.** This project investigates selected aspects of the earth move distance (EMD, [3]). As seen in class, computing the EMD reduces to solving a linear program (LP). In the sequel, the two sets of weighted points to be compared are denoted  $P = \{(p_i, w_i)\}_{i=1,\dots,m}$  and  $Q = \{(q_j, w'_j)\}_{j=1,\dots,n}$ .

#### Tasks.

1. In the class, we mentioned the property according to which the number of edges which carry flow is bounded by  $n + m - 1$ , with  $m$  and  $n$  the number of supply and demand vertices, respectively. Prove this claim.
2. EMD reduces to a linear program, and one easily finds a number of LP solvers, for example `lp_solve`. These solvers generally take as input a LP written in some standard format, e.g. MPS, see [https://en.wikipedia.org/wiki/MPS\\_%28format%29](https://en.wikipedia.org/wiki/MPS_%28format%29).  
One of them is `lp_solve`. Describe the algorithm used by `lp_solve`, and comment on its complexity. Is this complexity optimal?
3. Given the weighted points sets  $P$  and  $Q$ , write a program, in python or C++, writing the LP to a file, amenable to processing by `lp_solve`.
4. Generate random point sets  $P$  and  $Q$ . Then, run experiments by varying  $n$  and  $m$ . Record the running time. Are these consistent with the complexity of `lp_solve` ?
5. It has been shown that the EMD can be used to compare two clusterings, [4]. Propose an implementation of this algorithm.
6. Use it to compare clusterings obtained with k-means – you may use the implementation from <http://scikit-learn.org>, or from the SBL ([http://sbl.inria.fr/doc/Cluster\\_engines-user-manual.html](http://sbl.inria.fr/doc/Cluster_engines-user-manual.html)).
7. k-means is known to incur instabilities. In the case where the number of centers used is larger than the number of clusters, propose a strategy using the previous cluster comparison method to circumvent these instabilities.

**Contact.** Frederic Cazals: [frederic.cazals@inria.fr](mailto:frederic.cazals@inria.fr)

## 4 Circumventing distance concentration phenomena

**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 [5], 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 [5] to a different molecular data set, namely an ensemble of conformations of a protein model known as BLN69 [6]. 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.html](http://sbl.inria.fr/doc/Molecular_distances-user-manual.html).

### Tasks.

1. An ensemble of  $N \sim 10^6$  local minima of the BLN69 protein model can be found at <http://sbl.inria.fr/data-models>. This set is denoted  $\mathcal{S}$  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  $\mathcal{T}$  in the sequel.
2. We wish to analyze pairwise distances between selected conformations. Since  $N$  precludes using all pairs, propose two procedures to:
  - select a subset  $\mathcal{S}_1$  of  $n$  conformations by retaining the low energy conformations only. Hint: you may use topological persistence, see e.g. [7].
  - select a subset  $\mathcal{S}_2$  of  $n$  conformations maximizing the distances between the conformations selected. Hint: you may use the smart seeding procedure used in k-means.

Practically, you may take  $n$  in the range  $[10^3, 10^4]$ .

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 [5, Fig 1 (C)] for the sets  $\mathcal{S}_1$  and  $\mathcal{S}_2$ .
4. Analyse the distributions of pairwise distances for the sets  $\mathcal{S}_1$  and  $\mathcal{S}_2$ . You may proceed in 2 directions:
  - As in [5], check whether portions of the distribution correspond to distances between random points drawn according to a Gaussian distribution.
  - Following the distance concentration phenomenon studied in class, you may check whether some concentration phenomenon is observed.
5. Finally, following [5], use a sigmoid function to *tone down* non informative distances. Using MDS, compute the 2D embedding defined by the distance matrix. Use this embedding to project in 2D the local minima from the set  $\mathcal{T}$ . Compare to the embedding obtained for the first question.

**Contact.** Frederic Cazals: [frederic.cazals@inria.fr](mailto:frederic.cazals@inria.fr)

## 5 Multivariate two-sample tests

**Description.** The maximum mean discrepancy (MMD) is a powerful test statistic to compare two distributions  $p$  and  $q$  [8]. However, it critically depends on the choice of a *kernel width*, and this choice is non trivial in particular when multiple scales are present in the data [9]. The goal of this project is to explore alternative ways to compute suitable kernel widths.

### Tasks.

1. Consider a 2D point cloud consisting of a grid of Gaussians blobs, as in [9]. Using the ToMATo algorithm seen in class, and whose code can be downloaded from [http://geometrica.saclay.inria.fr/data/Steve.Oudot/clustering/ToMATo\\_code.tgz](http://geometrica.saclay.inria.fr/data/Steve.Oudot/clustering/ToMATo_code.tgz) or from [http://sbl.inria.fr/doc/Morse\\_theory\\_based\\_analyzer-user-manual.html](http://sbl.inria.fr/doc/Morse_theory_based_analyzer-user-manual.html), automate the tasks of clustering the data set. Note that using several persistence thresholds will yields nested clusters, for which the kernel width will be estimated as the median distance within points in a cluster.
2. Use the previous analysis to carry out an automatic kernel width calculation. Then, run tests on Gaussian blobs. You will report tables / plots for the type I and type II errors, and the experiments will be run by varying:
  - the eigenvalue ratio used to generate Gaussians blobs for the distribution  $q$  (when testing the type II error),
  - the persistence threshold used to define the clustering,
  - the dimension  $D$  of the data.
3. In class, we have studied a simple non parametric two sample test, namely the Wilcoxon Mann Whitney (U) test. Propose a novel multivariate TST performing first a random projection of the two point clouds along a random direction, and then computing the U test.
4. Test your procedure on the data used for MMD, and compare both.
5. As an improvement, one may apply the U test on a fixed (pre-defined) number of random projection – call it  $L$ . Test this idea by reporting the type I and type II errors, for various values of  $L$ , e.g.  $L = 5, 20$ . To control the type I error: is a Bonferroni-like correction needed?

**Contact.** Frederic Cazals: [frederic.cazals@inria.fr](mailto:frederic.cazals@inria.fr)



## 6 Detection of differences via feedback analysis

**Description.** In this project, we shall compare two methods studied during the class: two-sampled testing and feedback analysis.

The first method is the multivariate two-sample test (TST) MMD [8, 9], used to compare two distributions, that is  $H_0 : p = q$  versus  $H_1 : p \neq q$ . Matlab implementations of different MMD estimators and tests are provided by the authors<sup>1</sup>. Two versions of MMD are provided in the R package `kernlab` in the function `kmmd`.

The second method is the discrepancy localization strategy based on the Jensen-Shannon divergence (JSD) [10], whose implementation is provided in the SBL at [http://sbl.inria.fr/doc/Density\\_difference\\_based\\_clustering-user-manual.html](http://sbl.inria.fr/doc/Density_difference_based_clustering-user-manual.html). In short, this feedback method allows one to localize the discrepancy between two distributions, based on the following divergence:

$$\delta(z) \equiv D_{\text{KL}}(P(\cdot|z) \| P(\cdot)). \quad (1)$$

Summarizing, the TST delivers a binary information (accept/reject the null), while the feedback analysis delivers clusters contributing to the JS divergence. The goal of this project is to study the relationship between both pieces of information.

### Tasks.

1. In [10], the conditional probability estimation is carried out using a k-nearest neighbors based regressor. What is the rationale for doing so? What is the meaning of adaptation to the intrinsic dimension in this context [11]?
2. Data generation. In the following, we consider distributions  $p$  and  $q$  defined by a mixture of gaussians. We assume that a given distribution (and the associated sample) is parameterized by
  - the ambient dimension  $d$ ,
  - the number of gaussians  $N$ ,
  - a translation vector  $\tau$  used to translate the mean of each gaussian.

Write a procedure, e.g. using numpy, to generate such samples.

3. MMD and type I error. Propose a strategy to check whether MMD is a test of level  $\alpha$ . Then, present experiments on the gaussian data. Discuss the results.
4. TST versus feedback under the null hypothesis. Consider two distributions  $p$  and  $q$  with  $p = q$ . Let us assume that one is willing to define a TST using the JS divergence as test statistic. Present a method to do so, and run experiments on the data used for question 3. NB: a permutation test may be used.
5. TST versus feedback under the alternative. Consider now distributions  $p$  and  $q$  with  $p \neq q$ . Note that the *magnitude* of difference between  $p$  and  $q$  may be changed by playing with the aforementioned parameter  $\tau$ .
  - Run experiments to compare the variation of the test statistic of MMD, and that of the JS divergence returned by the feedback.
  - Comment on the respective variations observed, keeping in mind two key parameters of the methods, namely the kernel width used by MMD, and the number of neighbors for knn in the feedback.

**Contact.** Frederic Cazals: [frederic.cazals@inria.fr](mailto:frederic.cazals@inria.fr)

<sup>1</sup>Available at [www.gatsby.ucl.ac.uk/~gretton/mmd/mmd.htm](http://www.gatsby.ucl.ac.uk/~gretton/mmd/mmd.htm) and [www.gatsby.ucl.ac.uk/~gretton/adaptMMD/adaptMMD.htm](http://www.gatsby.ucl.ac.uk/~gretton/adaptMMD/adaptMMD.htm).

## 7 Mode-seeking for detecting metastable states in protein conformations

**Description.** The goal of this project is to analyze protein conformations using mode-seeking techniques, in order to detect metastable states and their proximity relations.

Relevant protein conformations can be generated in various ways by exploiting the molecular dynamics. For instance, one can simulate the protein folding process at small timescales. Each conformation then gives rise to a vector with  $3n$  coordinates, 3 per atom on the backbone ( $n$  atoms in total). One of the challenges is to understand how the conformations regroup themselves into clusters called metastable states, within which the probability of transition is high whereas it is low in-between. These states can then be fed to some stochastic process (such as a Markov chain) for efficient large timescale simulation. See [12] for more background.

The difficulty of recovering the metastable states stems from the fact that the clustering occurs in fairly high dimension ( $n$  can be of the order of the hundreds or thousands), with data that are not sampled along linear structures and clusters that are nonconvex. This is where mode-seeking techniques can help. Assuming the data points have been sampled iid from some unknown probability distribution, the principle of mode-seeking is to use an approximation of the gradient flow of the probability density function to push the data points towards the density maxima. These maxima then serve as cluster centers, and their preimages through the gradient flow are their corresponding clusters.

In this project we will use the topology-based method ToMaTo [13] to cluster the conformations. The goal is to get the same kind of results as in [12] and [13].

### Tasks.

1. Collect the data:
  - the set of alanine dipeptide conformations ([http://geometrica.saclay.inria.fr/data/Steve.Oudot/clustering\\_project/aladip\\_implicit.xyz](http://geometrica.saclay.inria.fr/data/Steve.Oudot/clustering_project/aladip_implicit.xyz)): 3 coordinates per atom, 10 atoms per conformation, 1 atom per line (so 10 lines per conformation, seen as a 30-dimensional point)
  - the set of conformations projected down to 2 dimensions for visualization ([http://geometrica.saclay.inria.fr/data/Steve.Oudot/clustering\\_project/dihedral.xyz](http://geometrica.saclay.inria.fr/data/Steve.Oudot/clustering_project/dihedral.xyz))
2. Compute the RMSD distance matrix between the 30-dimensional conformations (RMSD = Root Mean Square Deviation, see [https://en.wikipedia.org/wiki/Root-mean-square\\_deviation\\_of\\_atomic\\_positions](https://en.wikipedia.org/wiki/Root-mean-square_deviation_of_atomic_positions) for the definition, and <https://github.com/pandegroup/IRMSD> for some code to compute the RMSD).
3. Retrieve the code for ToMaTo at [http://geometrica.saclay.inria.fr/data/Steve.Oudot/clustering/ToMaTo\\_code.tgz](http://geometrica.saclay.inria.fr/data/Steve.Oudot/clustering/ToMaTo_code.tgz) and get familiar with it, e.g. try it out on the toy examples provided in the archive then play around with the parameters.
4. Try applying ToMaTo to the computed RMSD distance matrix. Beware that:
  - the code takes in a matrix of point coordinates and uses the Euclidean distance by default -i you should tweak the file Distance.h to your needs
  - the data are huge so the method will need some degree of optimization to scale up. Alternatively, you may want to consider applying it to subsamples of your data, although in that case you will need to find a mechanism to ascertain your results.
5. Hopefully you will be able to recover the same kind of result as in [12] and [13]. Don't forget to read these papers to get some insight into the data and their interpretation!

**Contact.** Steve Oudot: [steve.oudot@inria.fr](mailto:steve.oudot@inria.fr)

## 8 Analysis of the NBA postseason statistics using topological methods

**Description.** The aim here is to analyze the statistics of NBA players during the last season in order to gain insight into the playing styles of the players on the field. This is a typical exercise that data scientists do for club managers at the end of the regular season, to help them compose their team for the next season. The specificity of the project is to use topological techniques to perform the analysis. This has been done with success in the past, see e.g. [1]. Once the data have been collected and preprocessed (reorganized, renormalized, cleaned up), the idea will be to use a method called Mapper [14], which provides a higher-level understanding of the layout of the data than a mere clustering, and as such allows the user to make finer interpretations or predictions from the data. The students will be asked to reproduce at best the results obtained in [1], then to study their relevance and stability against perturbations of the data or parameters.

1. Collect the online data from the last NBA season. For instance, you can go to the CBS sports website (<http://www.cbssports.com/nba/stats/>) or to the NBA website (<http://stats.nba.com>).
2. Renormalize the data, select the variables and reduce the dimensionality. Inspect the layout of the data colored with labels (position types provided in the data set) in 2d and 3d. What do you observe?
3. Define a relevant metric on your data. For instance, as a weighted Euclidean metric. To set the weight, you can for instance:
  - perform classification (e.g. with single-linkage) on the data, and
  - optimize the weights according to some quality measure on the classification (e.g. Rand index since you have the true players' field positions at your disposal)
4. Get acquainted with the Mapper algorithm and test it against simple datasets. Then, apply Mapper to your data and optimize the parameters so as to recover a graph similar to the one from [15] (see <http://www.sloansportsconference.com/content/the-13-nba-positions-using-topology-to-identify-the-different-types-of-players/> for the video and slides from the talk)..
5. Result interpretation: inspect the subpopulations highlighted by the flares and loops in the graph, and compare them against the ones highlighted in [15] (beware that the players are different because the season is different). Are the results fully reproducible?
6. Study the stability of the resulting graph under perturbations of the data or Mapper parameters. What do you observe? What would be the best choices of parameters in terms of stability of the output? Would these choices be relevant on this data set?

**Software.** You can use: Python Mapper (<http://danifold.net/mapper/>), Kepler Mapper (<https://github.com/MLWave/kepler-mapper>)

**Contact.** Steve Oudot: [steve.oudot@inria.fr](mailto:steve.oudot@inria.fr)

## References

- [1] Peter N Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *ACM SODA*, volume 93, pages 311–321, 1993.
- [2] S. Vempala. Randomly-oriented kd trees adapt to intrinsic dimension. In *FSTTCS*, pages 48–57, 2012.
- [3] Y. Rubner, C. Tomasi, and L.J. Guibas. The earth mover’s distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.
- [4] D. Zhou, J. Li, and H. Zha. A new mallows distance based metric for comparing clusterings. In *ICML*, pages 1028–1035. ACM, 2005.
- [5] M. Ceriotti, G. Tribello, and M. Parrinello. Simplifying the representation of complex free-energy landscapes using sketch-map. *PNAS*, 108(32):13023–13028, 2011.
- [6] A. Roth, T. Dreyfus, C.H. Robert, and F. Cazals. Hybridizing rapidly growing random trees and basin hopping yields an improved exploration of energy landscapes. *J. of Computational Chemistry*, 37(8):739–752, 2016.
- [7] F. Cazals, T. Dreyfus, D. Mazauric, A. Roth, and C.H. Robert. Conformational ensembles and sampled energy landscapes: Analysis and comparison. *J. of Computational Chemistry*, 36(16):1213–1231, 2015.
- [8] A. Gretton, K.M. Borgwardt, J.R. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.
- [9] A. Gretton, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, K. Fukumizu, and B.K. Sriperumbudur. Optimal kernel choice for large-scale two-sample tests. In *Advances in Neural Information Processing Systems*, pages 1205–1213, 2012.
- [10] F. Cazals and A. Lhéritier. Beyond two-sample-tests: Localizing data discrepancies in high-dimensional spaces. In P. Gallinari, J. Kwok, G. Pasi, and O. Zaiane, editors, *IEEE/ACM International Conference on Data Science and Advanced Analytics*, Paris, 2015. Preprint: Inria tech report 8734.
- [11] Samory Kpotufe. k-nn regression adapts to local intrinsic dimension. In *Advances in Neural Information Processing Systems*, pages 729–737, 2011.
- [12] J. Chodera, W. Swope, J. Pitara, and K. Dill. Long-time protein folding dynamics from short-time molecular dynamics simulations. *Multiscale Modeling & Simulation*, 5(4):1214–1226, 2006.
- [13] F. Chazal, L. Guibas, S. Oudot, and P. Skraba. Persistence-based clustering in riemannian manifolds. *J. ACM*, 60(6):1–38, 2013.
- [14] G. Singh, F. Mémoli, and G. Carlsson. Topological methods for the analysis of high dimensional data sets and 3d object recognition. In *Symp. on Point Based Graphics*, pages 91–100, 2007.
- [15] M. Alagappan. From 5 to 13: Redefining the positions in basketball. In *MIT Sloan Sports Analytics Conference*, 2012.