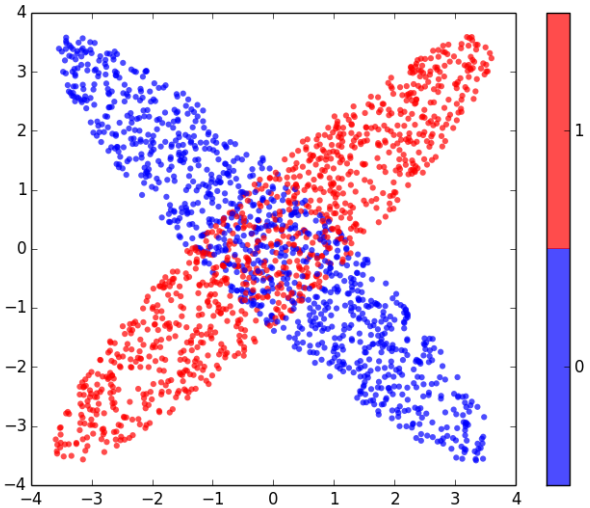
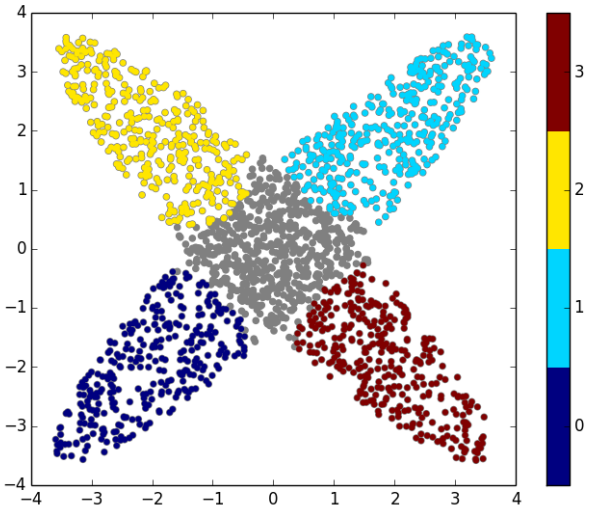
Beyond Two-sample-tests: Localizing Data Discrepancies in High-dimensional Spaces

Frederic.Cazals@inria.fr, Alix.Lheritier@inria.fr
Inria Sophia Antipolis, Algorithms-Biology-Structure

- `http://team.inria.fr/abs`
- `http://sbl.inria.fr`





Beyond two-sample tests

Data discrepancies: two-sample problem and effect size

- The two-sample test (TST) approach
  - Two datasets $x^{(n_0)} \equiv \{x_1, \ldots, x_{n_0}\}$ and $y^{(n_1)} \equiv \{y_1, \ldots, y_{n_1}\}$ in $\mathbb{R}^d$
    as i.i.d. samples from two unknown densities f_X and f_Y
  - Hypothesis testing:
    $\mathtt{H}_0 : f_X = f_Y \, a.e.,$
    $\mathtt{H}_1 : \neg \mathtt{H}_0$

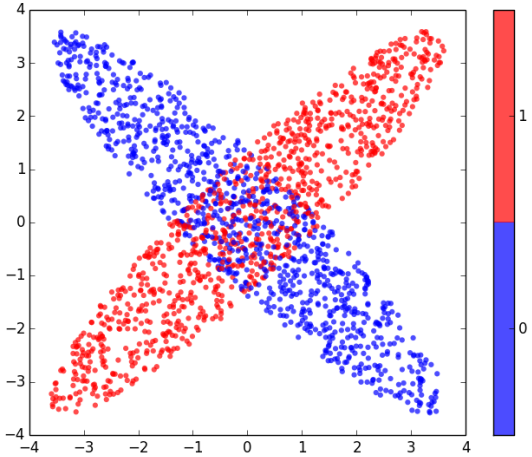
  $\rightarrow$ reject based on p-value: summarizes the difference in one bit!

- Effect size: "quantitative measure of the strength of a phenomenon"
  - p-value: magnitude of the statistical significance?
    for consistent TST: large sample size implies significance
  - effect size: various options for univariate data
    normalized difference between means

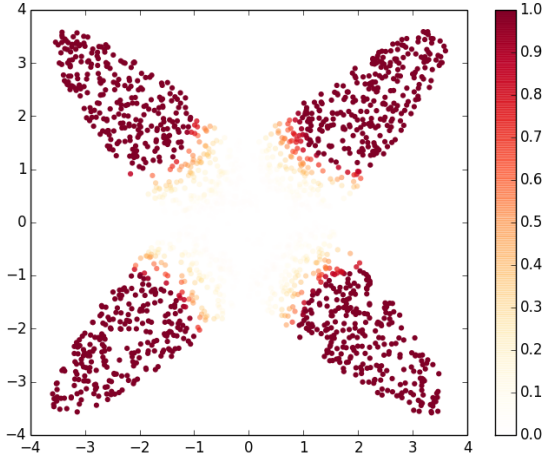
- Towards a notion of nonparametric multivariate effect size:
  - accommodating general discrepancies in $\mathbb{R}^d$
  - amenable to comparisons via some kind of normalization

What do we provide?

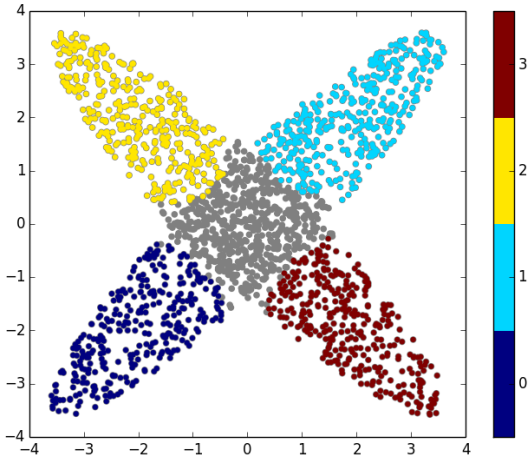
Comparing two point clouds:



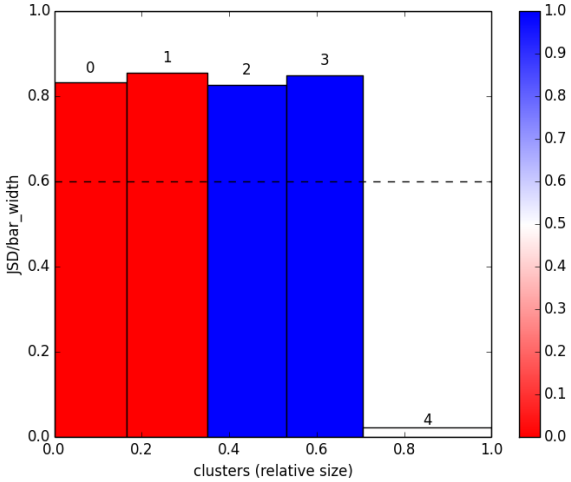
1. compute a local discrepancy,



2. find spatially coherent regions of high discrepancy,



3. provide a cluster based normalized effect size.



Outline of our method: three steps

- **Step 1:** Estimate a measure of local discrepancy at each given point
  - Ingredient: information theory
- **Step 2:** Aggregate local discrepancy in a spatially coherent way, to produce clusters by removing low discrepancy points
  - Ingredient: topological persistence
- **Step 3:** Produce an effect size barplot to summarize the discrepancy profile
- **Aftermath:** Assess the stability of clusters

Beyond two-sample tests

Pre-requisite: Jensen-Shannon divergence

▷ Kullback-Leibler divergence (KLD):

$$\begin{cases} D_{\mathrm{KL}}\left(f \| g\right) \equiv \int_{-\infty}^{\infty} f(x) \log \frac{f(x)}{g(x)} dx \\ \\ D_{\mathrm{KL}}\left(P \| Q\right) \equiv \sum_{l \in \mathcal{A}} P(l) \log \frac{P(l)}{Q(l)} \end{cases}$$

▷ The Jensen-Shannon divergence (JSD): symmetrizes and smoothes the KLD:
Consider $f \equiv (f_X + f_Y)/2$, then

$$JS\left(f_X \| f_Y\right) \equiv \frac{1}{2}\left(D_{\mathrm{KL}}\left(f_X \| f\right) + D_{\mathrm{KL}}\left(f_Y \| f\right)\right)$$

▷ Main properties of JS divergence:

- JSD is symmetric
- JSD is bounded between 0 and 1
- Its square root yields a metric

Step 1: Jensen-Shannon divergence and its decomposition

- Notations: two unknown densities f_X and f_Y, and the associated samples $x^{(n_0)}$ and $y^{(n_1)}$

- Define two random variables:
  - a position variable Z with density $f_Z \equiv f = (f_X + f_Y)/2$
  - a binary label $L \in \{0, 1\}$ with pmf $P(0) = 1/2$,
    indicating from which density (f_X or f_Y) an instance of Z is obtained.

- Equivalently, one defines a random vector:

$$(L, Z) = \begin{cases} (0, X) & \text{with prob. } \frac{1}{2} \\ (1, Y) & \text{with prob. } \frac{1}{2} \end{cases}$$

- Associated conditional and unconditional mass functions:

$$\begin{cases} P(l|z) = \mathbb{P}(L = l | Z = z) \\ P(l) = \mathbb{P}(L = l) = \frac{1}{2} \end{cases}$$

- Lemma: the JSD can be expressed as:

$$JS(f_X \| f_Y) = \int_{\mathbb{R}^d} f_Z(z) D_{\mathrm{KL}}(P(\cdot|z) \| P(\cdot))\, dz$$

Step 1: the local discrepancy

- From
$$JS(f_X \| f_Y) = \int_{\mathbb{R}^d} f_Z(z) D_{\mathrm{KL}}\left(P(\cdot|z) \| P(\cdot)\right) dz$$
- We define the *discrepancy* at location z as
$$\delta(z) \equiv D_{\mathrm{KL}}\left(P(\cdot|z) \| P(\cdot)\right).$$
- Observation:
  - $\delta(z) \in [0,1]$ and $\delta(z) = 0 \Leftrightarrow f_X(z) = f_Y(z)$.
    NB: $P(I) = 1/2$; logarithm in base 2
- Exploiting the discrepancy: $P(I)$ is known but $P(I|z)$ is not:

  we need to estimate $P(I|z)$ at each given location z.

Step 1: random design nonparametric regression

- Consider random variables: location $Z \in \mathbb{R}^d$, and response variable $R \in \mathbb{R}$
- Associated regression function:

$$m(z) \equiv \mathbb{E}\left[R|Z = z\right].$$

- Consider data: $\{(Z_i, R_i)\}_{i=1,\ldots,n}$
- k_n-nearest neighbor regressor: upon sorting samples by increasing distance to the query point z:

$$m_n(z) = \frac{1}{k_n} \sum_{i=1,\ldots,k_n} R_{(i,n)}(z)$$

▷Ref: L. Györfi and A. Krzyzak; A distribution-free theory of nonparametric regression; 2002

▷Ref: S. Kpotufe, NIPS 2011

Step 1: estimation via k-nearest neighbors

- Using the labels as response variable i.e. $R \equiv L$
- Using n i.i.d. realizations of (L, Z): build an estimator $m_n(z)$ for

$$m(z) = \mathbb{E}\left[L|Z = z\right] = P(1|z).$$

- Define the following estimator for $P(l|z)$: if $0 \leq m_n(z) \leq 1$:

$$\hat{P}_n\left(l|z\right) \equiv |1 - l - m_n(z)|.$$

- Thm: Using a k_n-nearest neighbor regressor, s.t. $\frac{k_n}{\log n} \to \infty$ and $\frac{k_n}{n} \to 0$:

$$\hat{\delta}_n(z) \equiv D_{\mathrm{KL}}\left(\hat{P}_n\left(\cdot|z\right)\|P(\cdot)\right) \xrightarrow{n\to\infty} \delta(z) \text{ a.s.}$$

for f-almost all $z \in \mathbb{R}^d$.

The random multiplexer to obtain i.i.d. realizations of (L, Z)

▷ A random sampler produces i.i.d. realizations of (Z, L) from $x^{(n_0)}$ and $y^{(n_1)}$:

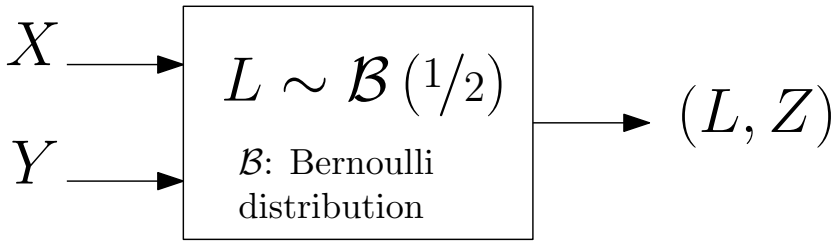


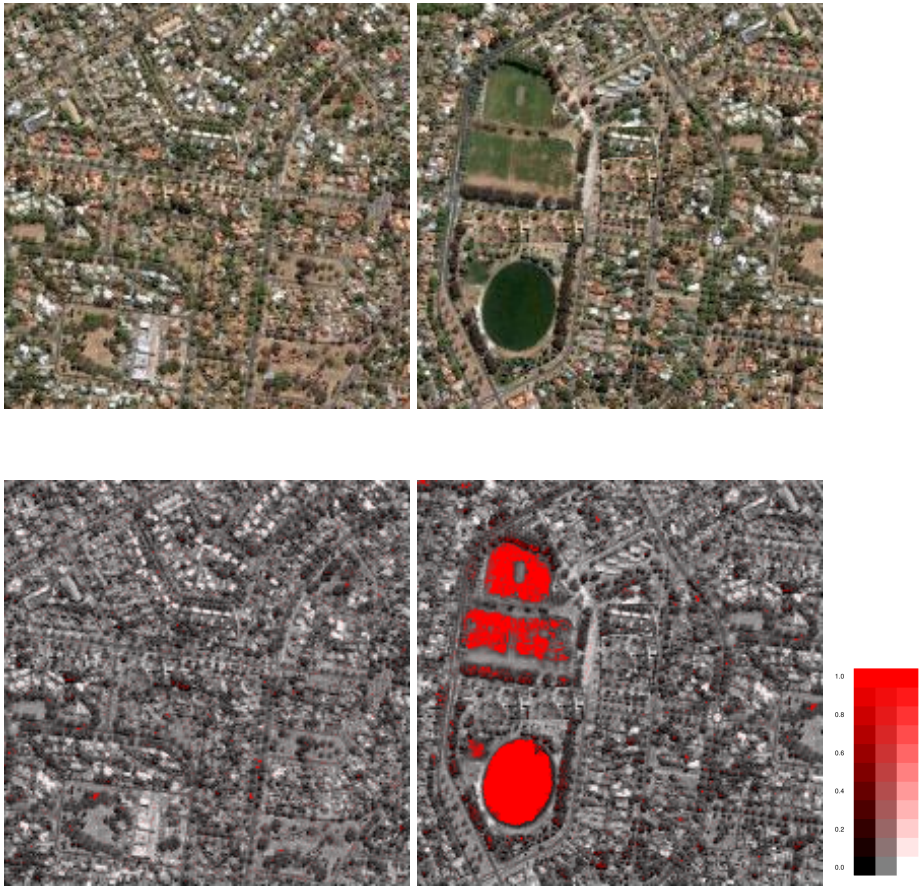
Figure: **Random multiplexer generating pairs (label, position).**

▷ The case of uneven populations:

– the multiplexer will consume faster the *small* population, and halt
– unused samples of the large population remain – detrimental information loss
– resample B times and take the median of estimates, on a per sample basis

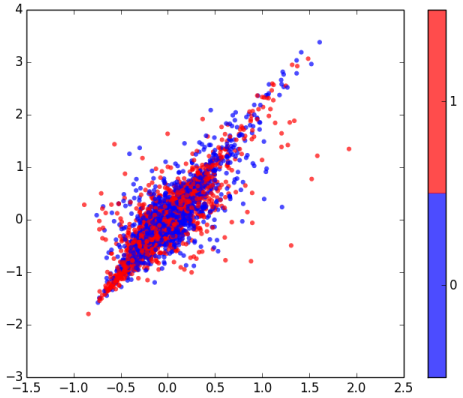
Step 1: Illustration: statistical image comparison

- ▷ Images: taking 2×2 blocks + (R,G,B) color coding: yields points in $\mathbb{R}^{12}$.
- ▷ Discrepancy estimate: using $k_n = n^{1/3}$
- ▷ Discrepancy plot: interpolate gray scale pixel color with red scale representing discrepancy at each pixel (upper left corner of the corresponding block)

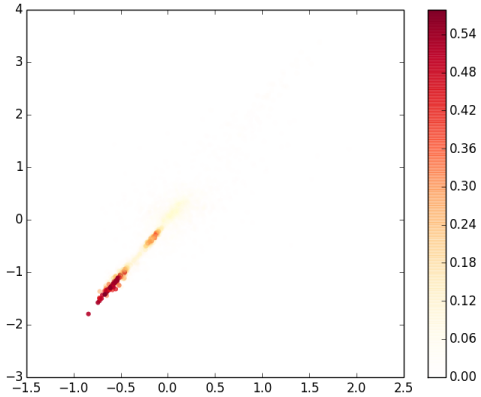


▷ Multidimensional Scaling of parameter space:

The two populations in $\mathbb{R}^2$...



...colored with $\hat{\delta}$:



Beyond two-sample tests

ToMATo: Topological Mode Analysis Tool

Persistence based clustering algorithm

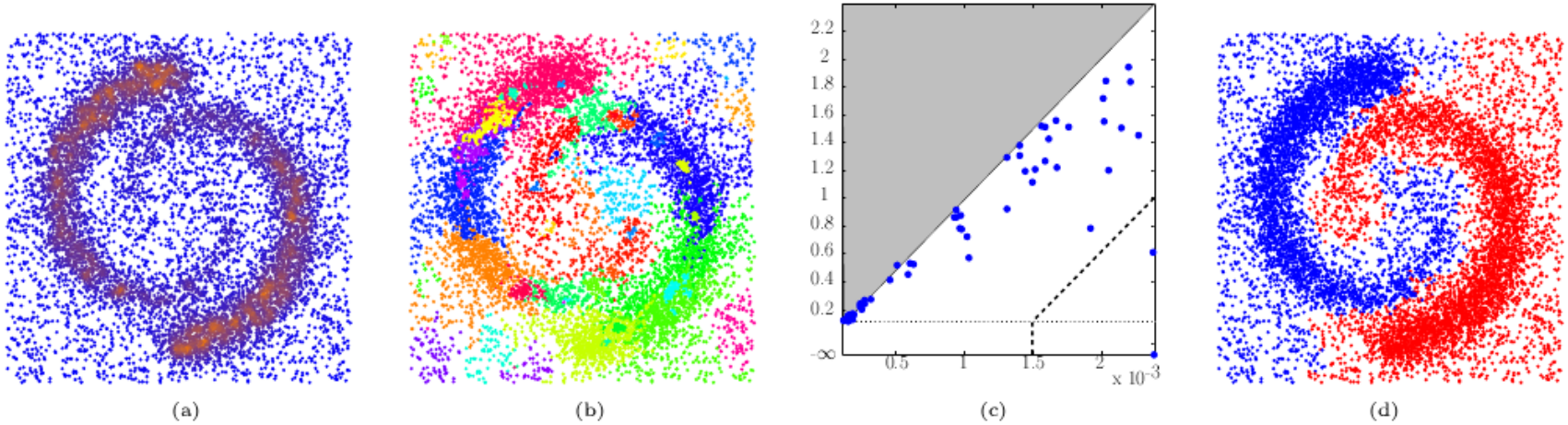
- Mode seeking strategy
  - Input: points sampled on a manifold
  - From a density estimation: height $\equiv$ estimated density
  - Find the persistent modes: one cluster per mode

    Step 1: build G^+

    Step 2: top-down processing yield potential merges between peaks
  - Key benefits

    existence diagram: estimate of the # of clusters

    works in a Riemannian setting—points on a manifold
- W.r.t. Morse theory: persistent maxima and their stable manifolds

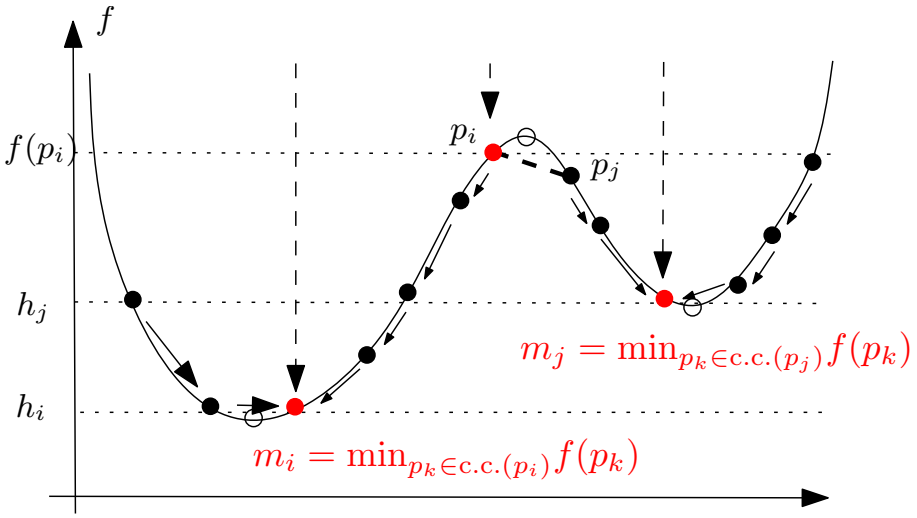


- Ref: Chazal, Guibas, Oudot, Skraba; ACM SoCG 2011

Persistent Minima and Sub-level Set Extraction from Samples: the Tomato Algorithm

- ▷ Input: NNG connecting samples on the landscape
- ▷ Output: DG + persistence diagram + one sub-level set
- ▷ Algorithm: relies on three operations at once, in 2 passes
- – quenches samples to their minima using a NNG discrete quench)
- – finds *bifurcations* i.e. pairs of sample across a *ridge*
- – cancels *non persistent basins* on the fly (Union-Find algorithm)

- – Variant if all samples have been quenched:
  detection of adjacencies between basins (yet: overestimation of barriers)

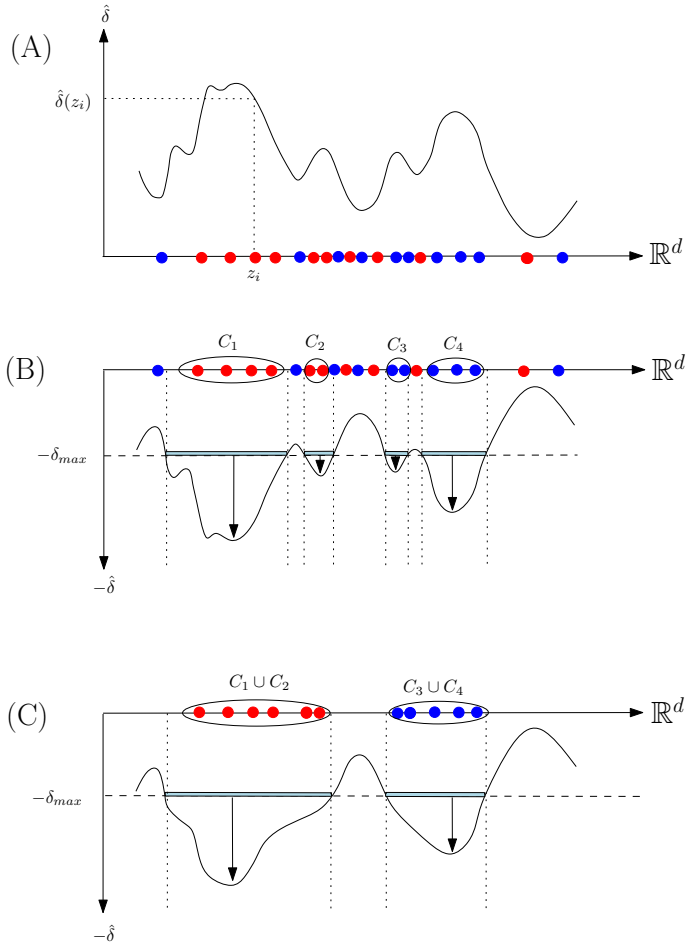


▷Ref: Chazal, Guibas, Oudot, Skraba; ACM SoCG 2011

Beyond two-sample tests

Step 2: Building the clusters from sublevel sets of $-\hat{\delta}(z)$

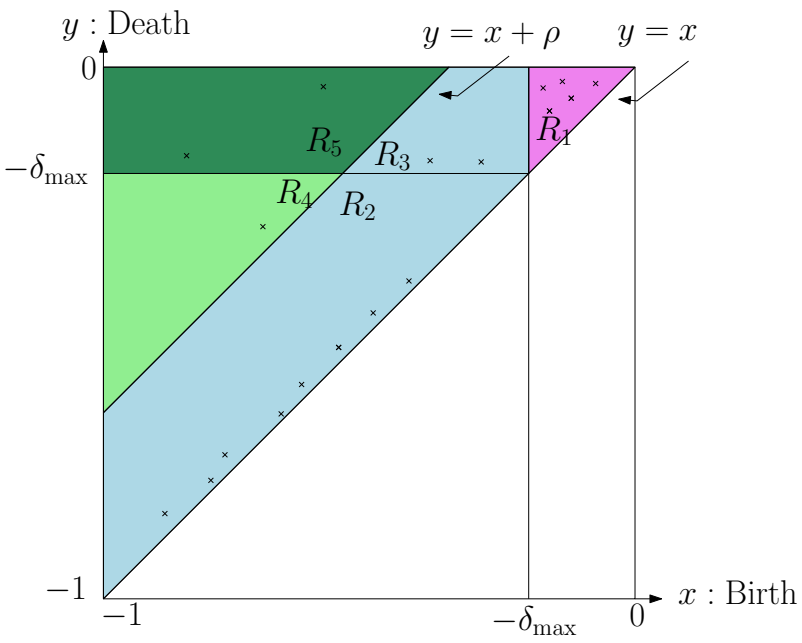
- Ingredients:
  - Height function $-\hat{\delta}(z)$ modeled with nearest neighbor graph
  - Parameter: discrepancy/significance threshold δ_{max}
- Construction:
  - Idea: one cluster $\sim$ one connected component of the sublevel set of $-\hat{\delta}(z)$ defined by δ_{max}
  - Extra ingredient: smoothing the landscape to get rid of small clusters : smoothing using topological persistence at threshold ρ



Step 2: Building the clusters: persistence diagram

▷ Partition of the PD induced by:

- Significance threshold δ_{max}
- Persistence threshold ρ



▷ Local minimum m of $-\hat{\delta}(z)$:

- Selected/rejected: m was born before $-\delta_{max}$.
- Persistent/canceled: persistence$(m) \geq \rho$
- Filtered (un-filtered): the catchment basin of m dies after (before) $-\delta_{max}$.

▷ Observation:

- # clusters : 1 + # points in region R_5 of the PD.
- # persistent local minima : 1 + num points in the region $R_4 \cup R_5$ of the PD.

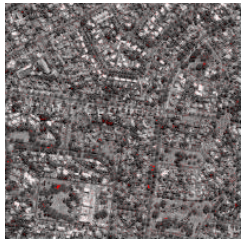
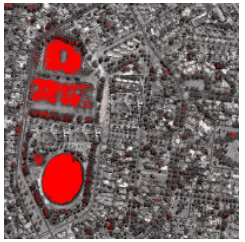
▷Ref: Chazal, Guibas, Oudot, Skraba, J. ACM, 2013

▷Ref: Cazals and D. Cohen-Steiner, CGTA, 2011

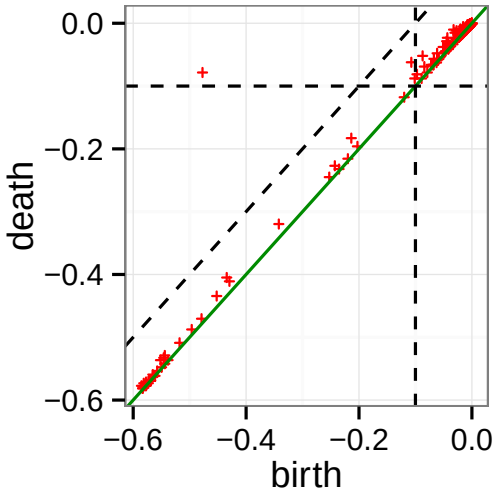
Step 2: Illustration: statistical image comparison

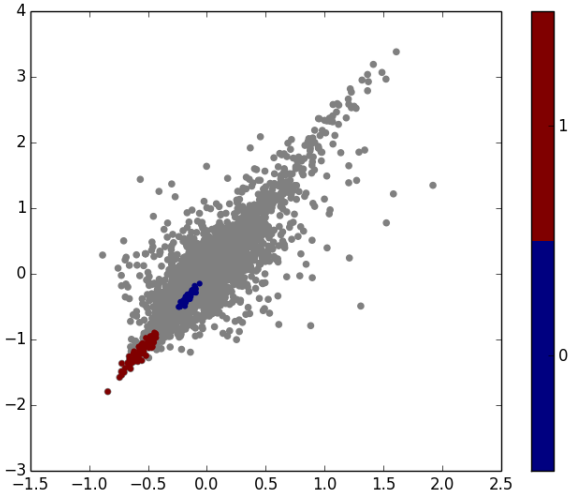
- Images again:



- Parameters: $k = 10$ (NNG), $\rho = 0.1, \delta_{max} = 0.1$





Beyond two-sample tests

Step 3: Effect size: discrepancy profile

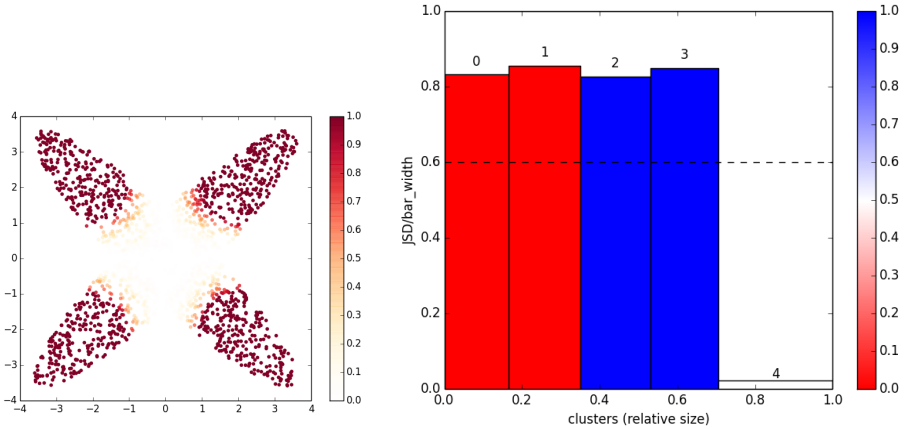
- Global estimated JSD: area under dashed line
- Maximum JSD: area under continuous line (=1)
- Contribution of each cluster C to JSD: area of bar

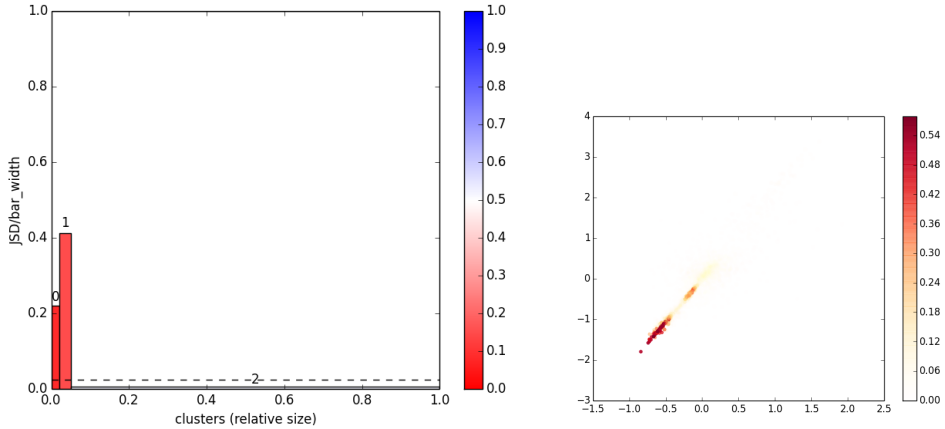
$$JS_C\left(f_X \| f_Y\right) \equiv \frac{1}{n_0 + n_1} \sum_{z \in (x^{(n_0)} \cup y^{(n_1)}) \cap C} \hat{\delta}(z).$$

- Mass of each cluster: bar width
- Population balance in each cluster: bar color (heat map)

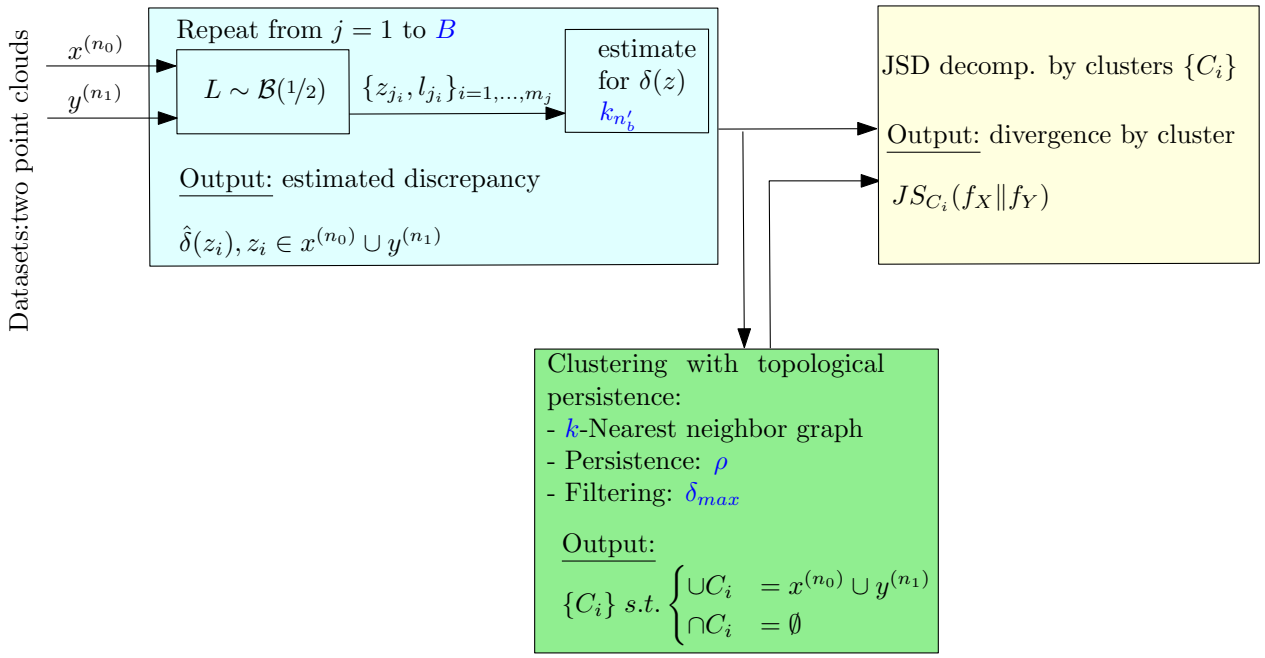
- Ellipses:
  - Large global JSD (dashed line)
  - Contributed by 2+2 balanced clusters

- Images:
  - Smaller global JSD (dashed line)
  - Contributed by 2 clusters





Wrapping-up: workflow

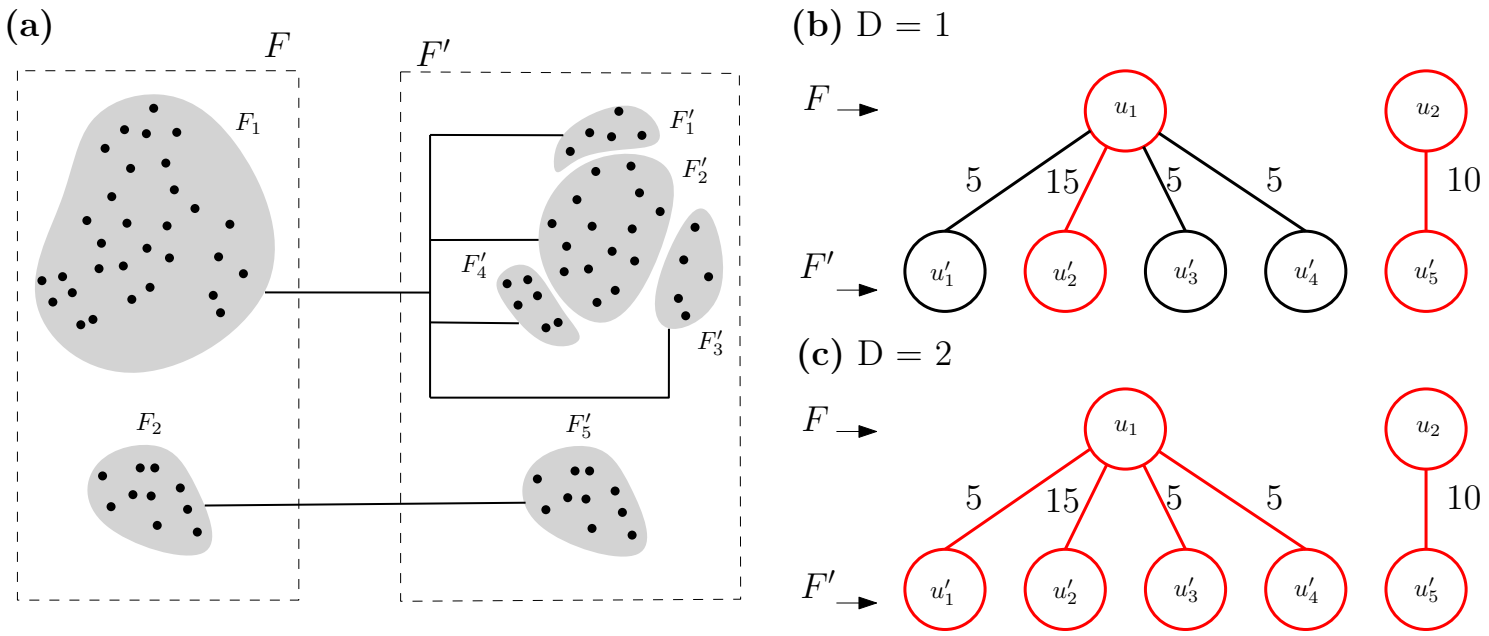


- ▷ Compulsory parameters:
  - k_n: regression parameter
  - δ_{max}: discrepancy significance threshold
  - ρ: persistence threshold
  - k: number of nearest neighbors for the persistence analysis
- ▷ Optional parameter:
  - B: num. repetition in case of unbalanced populations

Beyond two-sample tests

On the stability of clusterings

- Question: are the clusters stable w.r.t. these compulsory parameters?



- General approach: comparing clusterings via clusters of clusters
  - Find a matching between clusters of clusters, called meta-clusters
  - Controlled by a parameter D monitoring the diameter of meta-clusters
    In the example: compare $D = 1$ vs $D = 2$

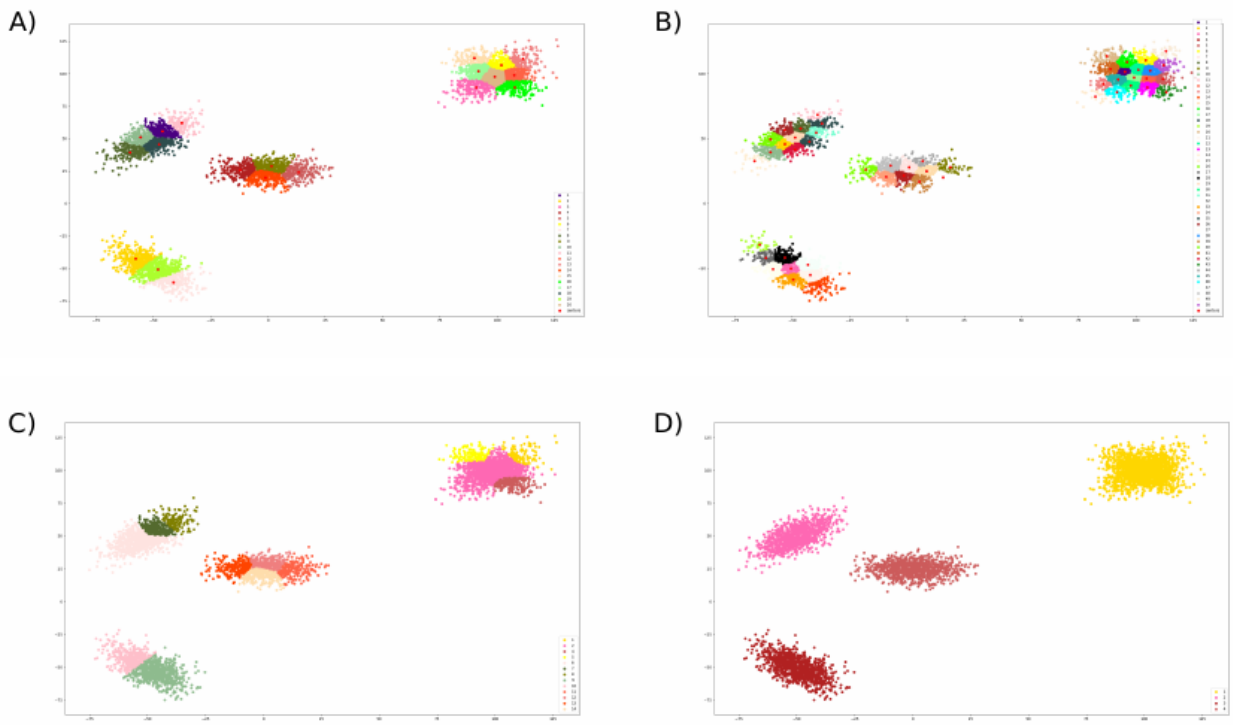
Comparing clusterings: at which *scale* do clusters merge?

What is the *right* number of clusters?

- ▷ Example:
  - Using `k-means++` to cluster 5000 samples from five Gaussian blobs
  - Using D-family matching to infer the *right/natural* # of clusters

(A) k-means++, $k = 20$

(B) k-means++, $k = 50$



(C) $D = 3$, 17 meta clusters, $\Phi_{=}(4)068$

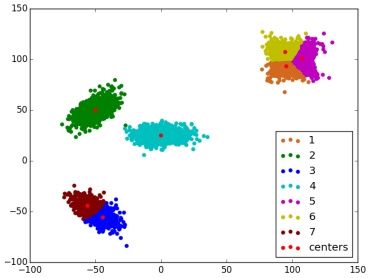
(D) $D = 4$, 4 meta clusters, $\Phi_{=}(5)000$

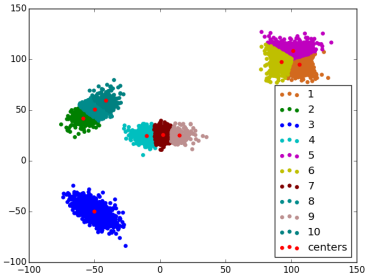
Comparing clusterings using matchings between clusters of clusters

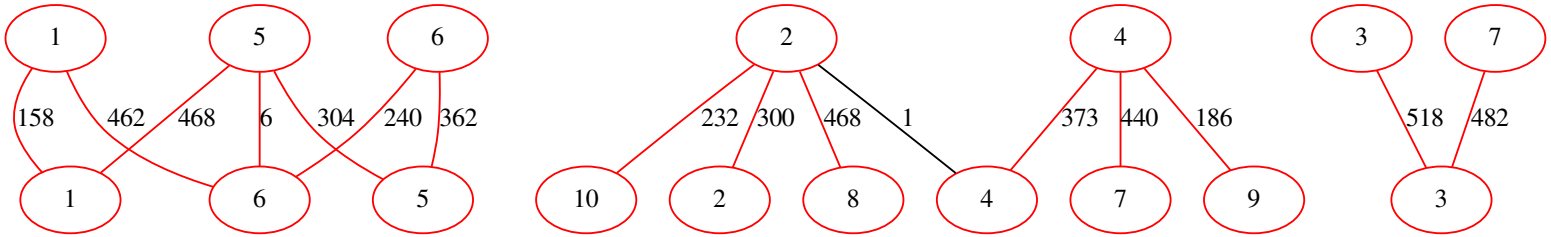
▷ Contributions:

– Formalization of the D-family matching problem
– NP-completeness results and unbounded approximation ratio for simple strategies
  Open: is the problem APX hard?
– Exact polynomial time algos. for selected intersection graphs (trees)
– Heuristics for general graphs
– Extensive experiments (vs. the variation of information)

▷ Stability of kmeans++:







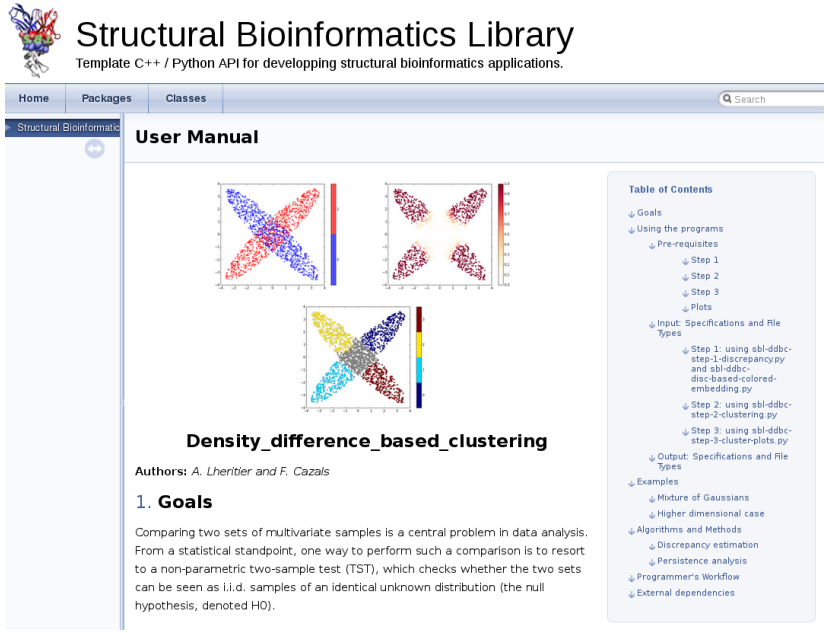
▷Ref: Cazals, Mazauric, Tetley, Watrigant; submitted

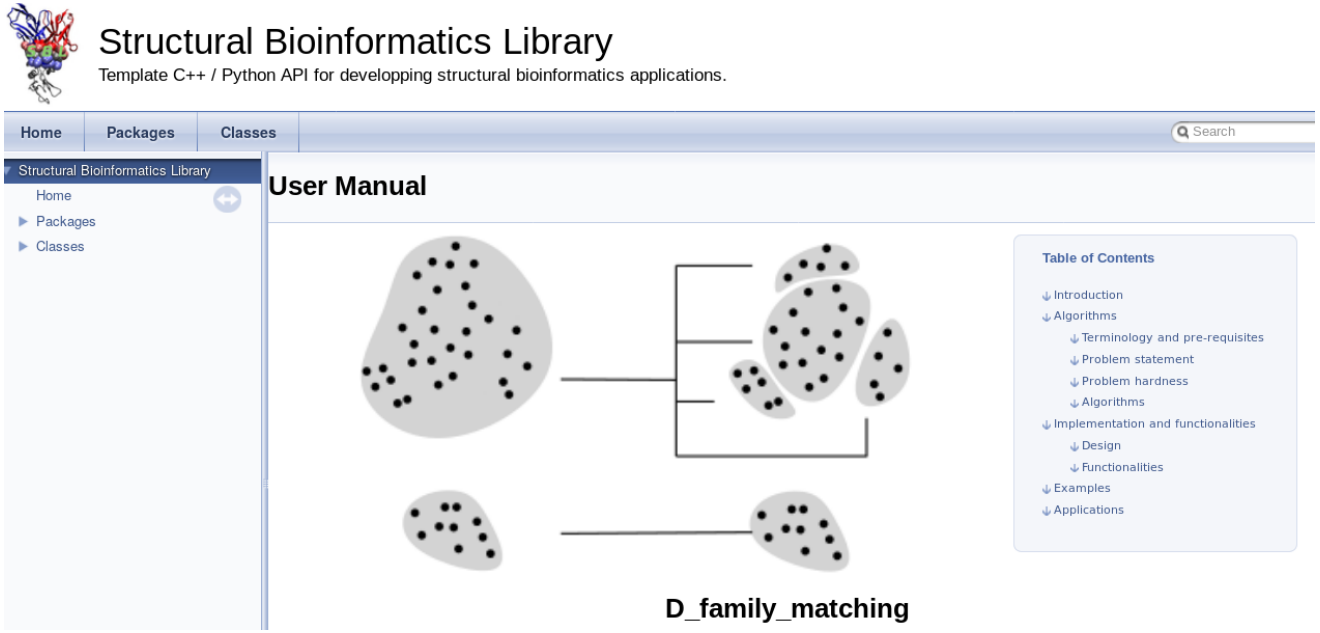
Beyond two-sample tests

Conclusions - Outlook

- An elementary method providing a normalized discrepancy based upon (provably correct) estimates of the JS divergence computed on a per point basis
- Merely requires an efficient algorithm for (approximate) nearest neighbors
- By changing the sign of the discreancy: can be used to find clusters of low discrepancy i.e. *coherent regions*
- Can be used as goodness-of-fit tool, by sampling from a given model, then comparing data to spot discrepancies
- Clusters can be post-processed separately: e.g., PCA to find relevant directions
    - See also Mueller and Jaakkola, NIPS 2015

Try me: http://sbl.inria.fr





Bibliography

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.
Inria tech report 8734.

A. Lhéritier and F. Cazals.
A sequential non-parametric two-sample test.
IEEE Transactions on Information Theory, NA, 2018.
Inria tech report 8704.

F. Cazals, D. Mazauric, R. Tetley, and R. Watrigant.
Comparing two clusterings using matchings between clusters of clusters.
Submitted, 2017.
Inria tech report 9063.

F. Cazals and T. Dreyfus.
The Structural Bioinformatics Library: modeling in biomolecular science and beyond.
Bioinformatics, 7(33):1–8, 2017.