entropy

Technical Note

Scikit-Dimension: A Python Package for Intrinsic Dimension Estimation

Jonathan Bac [1,2,3,*], Evgeny M. Mirkes [4,5], Alexander N. Gorban [4,5], Ivan Tyukin [4,5] and Andrei Zinovyev [1,2,3,5,*]

[1] Institut Curie, PSL Research University, 75248 Paris, France
[2] INSERM, U900, 75248 Paris, France
[3] CBIO-Centre for Computational Biology, Mines ParisTech, PSL Research University, 75272 Paris, France
[4] Department of Mathematics, University of Leicester, Leicester LE1 7RH, UK; em322@leicester.ac.uk (E.M.M.); a.n.gorban@leicester.ac.uk (A.N.G.); i.tyukin@leicester.ac.uk (I.T.)
[5] Laboratory of Advanced Methods for High-Dimensional Data Analysis, Lobachevsky University, 603105 Nizhniy Novgorod, Russia
[*] Correspondence: jonathan.bac@cri-paris.org (J.B.); andrei.zinovyev@curie.fr (A.Z.); Tel.: +33-156-246-989 (A.Z.)

Abstract: Dealing with uncertainty in applications of machine learning to real-life data critically depends on the knowledge of intrinsic dimensionality (ID). A number of methods have been suggested for the purpose of estimating ID, but no standard package to easily apply them one by one or all at once has been implemented in Python. This technical note introduces `scikit-dimension`, an open-source Python package for intrinsic dimension estimation. The `scikit-dimension` package provides a uniform implementation of most of the known ID estimators based on the scikit-learn application programming interface to evaluate the global and local intrinsic dimension, as well as generators of synthetic toy and benchmark datasets widespread in the literature. The package is developed with tools assessing the code quality, coverage, unit testing and continuous integration. We briefly describe the package and demonstrate its use in a large-scale (more than 500 datasets) benchmarking of methods for ID estimation for real-life and synthetic data.

Keywords: intrinsic dimension; effective dimension; Python package; method benchmarking

Citation: Bac, J.; Mirkes, E.M.; Gorban A.N.; Tyukin, I.; Zinovyev, A. Scikit-Dimension: A Python Package for Intrinsic Dimension Estimation. *Entropy* **2021**, *23*, 1368. https://doi.org/10.3390/e23101368

Academic Editor: Philip Broadbridge

Received: 17 September 2021
Accepted: 16 October 2021
Published: 19 October 2021

1. Introduction

We present `scikit-dimension`, an open-source Python package for global and local intrinsic dimension (ID) estimation. The package has two main objectives: (i) foster research in ID estimation by providing code to benchmark algorithms and a platform to share algorithms; and (ii) democratize the use of ID estimation by providing user-friendly implementations of algorithms using the scikit-learn application programming interface (API) [1].

ID intuitively refers to the minimum number of parameters required to represent a dataset with satisfactory accuracy. The meaning of "accuracy" can be different among various approaches. ID can be more precisely defined to be n if the data lie closely to a n-dimensional manifold embedded in R^d with little information loss, which corresponds to the so-called "manifold hypothesis" [2,3]. ID can be, however, defined without assuming the existence of a data manifold. In this case, data point cloud characteristics (e.g., linear separability or pattern of covariance) are compared to a model n-dimensional distribution (e.g., uniformly sampled n-sphere or n-dimensional isotropic Gaussian distribution), and the term "effective dimensionality" is sometimes used instead of "intrinsic dimensionality" as such n giving the most similar characteristics to the one measured in the studied point cloud [4,5]. In `scikit-dimension`, these two notions are not distinguished.

The knowledge of ID is important to determine the choice of machine learning algorithm, anticipate the uncertainty of its predictions, and estimate the number of sufficiently distinct clusters of variables [6,7]. The well-known *curse of dimensionality*, which states that many problems become exponentially difficult in high dimensions, does not depend on the number of features, but on the dataset's ID [8]. More precisely, the effects of the dimensionality curse are expected to be manifested when $ID \gg ln(M)$, where M is the number of data points [9,10].

Current ID estimators have diverse operating principles (we refer the reader to [11] for an overview). Each ID estimator is developed based on a selected feature (such as the number of data points in a sphere of fixed radius, linear separability or expected normalized distance to the closest neighbor), which scales with n: therefore, various ID estimation methods provide different ID values. Each dataset can be characterized by a unique *dimensionality profile* of ID estimations, according to different existing methods, which can serve as an important signature for choosing the most appropriate data analysis method.

Dimensionality estimators that provide a single ID value for the whole dataset belong to the category of global estimators. However, datasets can have complex organizations and contain regions with varying dimensionality [9]. In such a case, they can be explored using local estimators, which estimate ID in local neighborhoods around each point. The neighborhoods are typically defined by considering the k closest neighbors. Such approaches also allow repurposing global estimators as local estimators.

The idea behind local ID estimation is to operate at a scale where the data manifold can be approximated by its tangent space [12]. In practice, ID is sensitive to scale, and choosing the neighborhood size is a trade-off between opposite requirements [11,13]: ideally, the neighborhood should be big relative to the scale of the noise, and contain enough points. At the same time, it should be small enough to be well approximated by a flat and uniform tangent space.

We perform benchmarking of 19 ID estimators on a large collection of real-life and synthetic datasets. Previously, estimators were benchmarked based mainly on artificial datasets representing uniformly sampled manifolds with known ID [4,11,14], comparing them for the ability to estimate the ID value correctly. Several ID estimators were used on real-life datasets to evaluate the degree of dimensionality curse in a study of various metrics in data space [15]. Here, we benchmark ID estimation methods, focusing on their applicability to a wide range of datasets of different origin, configuration and size. We also look at how different ID estimations are correlated, and show how `scikit-dimension` can be used to derive a consensus measure of data dimensionality by averaging multiple individual measures. The latter can be a robust measure of data dimensionality in various applications.

`Scikit-dimension` was applied in several recent studies for estimating the intrinsic dimensionality of real-life datasets [16,17].

2. Materials and Methods

2.1. Software Features

`Scikit-dimension` is an open-source software available at https://github.com/j-bac/scikit-dimension (accessed on 18 October 2021).

`Scikit-dimension` consists of two modules. The *id* module provides ID estimators, and the *datasets* module provides synthetic benchmark datasets.

2.1.1. *id* Module

The *id* module contains estimators based on the following:

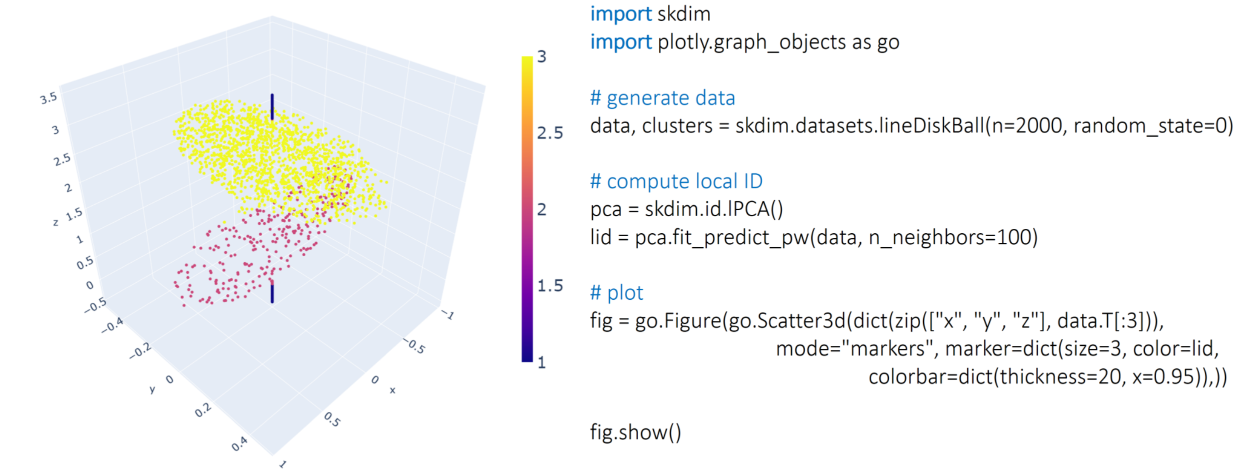
- Correlation (fractal) dimension (id.CorrInt) [18].
- Manifold-adaptive fractal dimension (id.MADA) [19].
- Method of moments (id.MOM) [20].
- Principal component analysis (id.lPCA) [3,21–23].
- Maximum likelihood (id.MLE) [24–26].
- Minimum spanning trees (id.KNN) [27].

- Estimators based on concentration of measure (id.MiND_ML, id.DANCo, id.ESS, id.TwoNN, id.FisherS, id.TLE) [4,28–33].

The description of the method principles is provided together with the package documentation at https://scikit-dimension.readthedocs.io/ (accessed on 18 October 2021) and in reviews [5,9,14].

2.1.2. *Datasets* Module

The *datasets* module allows user to test estimators on synthetic datasets; Figure 1. It can generate several low-dimensional toy datasets to play with different estimators as well as a set of synthetic manifolds commonly used to benchmark ID estimators, introduced by [14] and further extended in [11,28].



```
import skdim
import plotly.graph_objects as go

# generate data
data, clusters = skdim.datasets.lineDiskBall(n=2000, random_state=0)

# compute local ID
pca = skdim.id.lPCA()
lid = pca.fit_predict_pw(data, n_neighbors=100)

# plot
fig = go.Figure(go.Scatter3d(dict(zip(["x", "y", "z"], data.T[:3])),
                       mode="markers", marker=dict(size=3, color=lid,
                                      colorbar=dict(thickness=20, x=0.95)),))

fig.show()
```

Figure 1. Example usage: generating the Line–Disk–Ball dataset [10]), which has clusters of varying local ID, and coloring points by estimates of local ID obtained by id.lPCA.

2.2. *Development*

`Scikit-dimension` is built according to the scikit-learn API [1] with support for Linux, MacOS, Windows and Python >= 3.6. The code style and API design are based on the guidelines of scikit-learn, with the NumPy [34] documentation format, and continuous integration on all three platforms. The online documentation is built using Sphinx and hosted with ReadTheDocs.

2.3. *Dependencies*

`Scikit-dimension` depends on a limited number of external dependencies on the user side for ease of installation and maintenance:

- Matplotlib [35]
- Pandas [36].
- Scikit-learn [1].
- Numba [37].
- SciPy [38]
- NumPy [34].

2.4. *Related Software*

Related open-source software for ID estimation have previously been developed in different languages such as R, MATLAB or C++ and contributed to the development of `scikit-dimension`.

In particular, refs. [10,39–41] provided extensive collections of ID estimators and datasets for R users, with [40] additionally focusing on dimension reduction algorithms.

Similar resources can be found for MATLAB users [42–45]. Benchmarking many of the methods for ID estimation included in this package was performed in [15]. Finally, there exist several packages implementing standalone algorithms; in particular for Python, we refer the reader to complementary implementations of the GeoMLE, full correlation dimension, and GraphDistancesID algorithms [46–49].

To our knowledge, `scikit-dimension` is the first Python implementation of an extensive collection of ID methods. Compared to similar efforts in other languages, the package puts emphasis on estimators, quantifying various properties of high-dimensional data geometry, such as the concentration of measure. It is the only package to include ID estimation based on linear separability of data, using Fisher discriminants [4,32,50,51].

3. Results

3.1. Benchmarking `Scikit-Dimension` on a Large Collection of Datasets

In order to demonstrate the applicability of `scikit-dimension` to a wide range of real-life datasets of various configurations and sizes, we performed a large-scale benchmarking of `scikit-dimension`, using the collection of datasets from the `OpenML` repository [52]. We selected those datasets having at least 1000 observations and 10 features, without missing values. We excluded those datasets which were difficult to fetch, either because of their size or an error in the `OpenML` API. After filtering out repetitive entries, 499 datasets were collected. Their number of observations varied from 1002 to 9,199,930, and their number of features varied from 10 to 13,196. We focused only on numerical variables, and we subsampled the number of rows in the matrix to a maximum of 100,000. All dataset features were scaled to unit interval using Min/Max scaling. In addition, we filtered out approximate non-unique columns and rows in the data matrices since some of the ID methods could be affected by the presence of identical (or approximately identical) rows or columns.

We added to the collection 18 datasets, containing single-cell transcriptomic measurements, from the CytoTRACE study [53] and 4 largest datasets from The Cancer Genome Atlas (TCGA), containing bulk transcriptomic measurements. Therefore, our final collection contained 521 datasets altogether.

3.1.1. `Scikit-Dimension` ID Estimator Method Features

We systematically applied 19 ID estimation methods from `scikit-dimension`, with default parameter values, including 7 methods based on application of principal component analysis (“linear” or PCA-based ID methods), and 12 based on application of various other principles, including correlation dimension and concentration of measure-based methods (“nonlinear” ID methods).

For KNN and MADA methods, we had to further subsample the data matrix to a maximum of 20,000 rows; otherwise they were too greedy in terms of memory consumption. Moreover, DANCo and ESS methods appeared to be too slow, especially in the case of a large number of variables: therefore, we made ID estimations in these cases on small fragments of data matrices. Thus, for DANCo, the maximum matrix size was set to 10,000 × 100, and for ESS to 2000 × 20. The number of features was reduced for these methods when needed, by using PCA-derived coordinates, and the number of observations were reduced by random subsampling.

In Table 1, we provide the summary of characteristics of the tested methods. In more detail, the following method features were evaluated (see Figure 2).

Firstly, we simply looked at the ranges of ID values produced by the methods across all the datasets. These ranges varied significantly between the methods, especially for the linear ones (Figure 2A).

Secondly, we tested the methods with respect to their ability to successfully compute the ID as a positive finite value. It appeared that certain methods (such as MADA and TLE), in a certain number of cases produced a significant fraction of uninterpretable estimates (such as “nan” or negative value); Figure 2B. We assume that in most of such cases, the problem with ID estimation is caused by the method implementation, not anticipating certain relatively rare data point configurations, rather than the methodology itself, and that a reasonable ID estimate always exists. Therefore, in the case of an uninterpretable value due to method implementation, for further analysis, we considered it possible to impute the ID value from the results of application of other methods; see below.

Thirdly, for a small number of datasets, we performed a test of their sensitivity to the presence of strongly redundant features. For this purpose, we duplicated all features in a matrix and recomputed the ID. The resulting sensitivity is the ratio between the ID computed for the larger matrix and the ID computed for the initial matrix, having no duplicated columns. It appears that despite most of the methods being robust with respect to such a matrix duplication, some (such as PCA-based broken stick or the famous Kaiser methods popular in various fields, such as biology [54,55]), tend to be very sensitive (Figure 2C), which is compliant with some previous reports [15].

Table 1. Summary table of ID methods characteristics. The qualitative score changes from "− − −" (worst) to "+++" (best).

Method Name	Short Name(s)	Ref(s)	Valid Result	Insensitivity to Redundancy	Uniform ID Estimate in Similar Datasets	Performance with Many Observations	Performance with Many Features
PCA Fukunaga-Olsen	PCA FO, PFO	[15,22]	+++	+++	+++	+++	+++
PCA Fan	PFN	[23]	+++	+++	+++	+++	+++
PCA maxgap	PMG	[56]	+++	− − −	+	+++	+++
PCA ratio	PRT	[57]	+++	+++	+	+++	+++
PCA participation ratio	PPR	[57]	+++	+++	++	+++	+++
PCA Kaiser	PKS	[54,58]	+++	−	+++	+++	+++
PCA broken stick	PBS	[55,59]	+++	− −	+++	+++	+++
Correlation (fractal) dimensionality	CorrInt, CID	[18]	+	+++	++	+	+
Fisher separability	FisherS, FSH	[4,32]	++	+++	+++	++	+++
K-nearest neighbours	KNN	[27]	++	− −	− −	−	++
Manifold-adaptive fractal dimension	MADA, MDA	[19]	−	+++	+++	−	+
Minimum neighbor distance—ML	MIND_ML, MMk, MMi	[28]	+++	+++	++	++	+
Maximum likelihood	MLE	[25]	++	+++	++	++	+
Methods of moments	MOM	[20]	+++	+++	+++	++	+
Estimation within tight localities	TLE	[33]	− −	+++	+++	++	+
Minimal neighborhood information	TwoNN, TNN	[31]	++	+++	+++	++	+++
Angle and norm concentration	DANCo, DNC	[29]	+	+++	+++	− − −	− − −
Expected simplex skewness	ESS	[56]	+++	+++	+++	− − −	− − −

Some of the datasets in our collection could be combined in homogeneous groups according to their origin, such as the data coming from quantitative structure–activity relationship (QSAR)–based quantification of a set of chemicals. The size of the QSAR fingerprint for the molecules is the same in all such datasets (1024 features): therefore, we can assume that the estimate of ID will not vary too much across the datasets from the same group. We computed the coefficient of a variation of ID estimates across three such dataset groups, which revealed that certain methods tend to provide less stable estimations than the others; Figure 2D.

Finally, we recorded the computational time needed for each method. We found that the computational time could be estimated with good precision ($R^2 > 0.93$ for all ID estimators), using the multiplicative model: $Time = c \times N_{obj}^{\alpha} \times N_{var}^{\beta}$, where N_{obj} and N_{var} are the number of objects and features in a dataset, correspondingly. Using this model fit for each method, we estimated the time needed to estimate ID for data matrices of four characteristic sizes; Figure 2E.

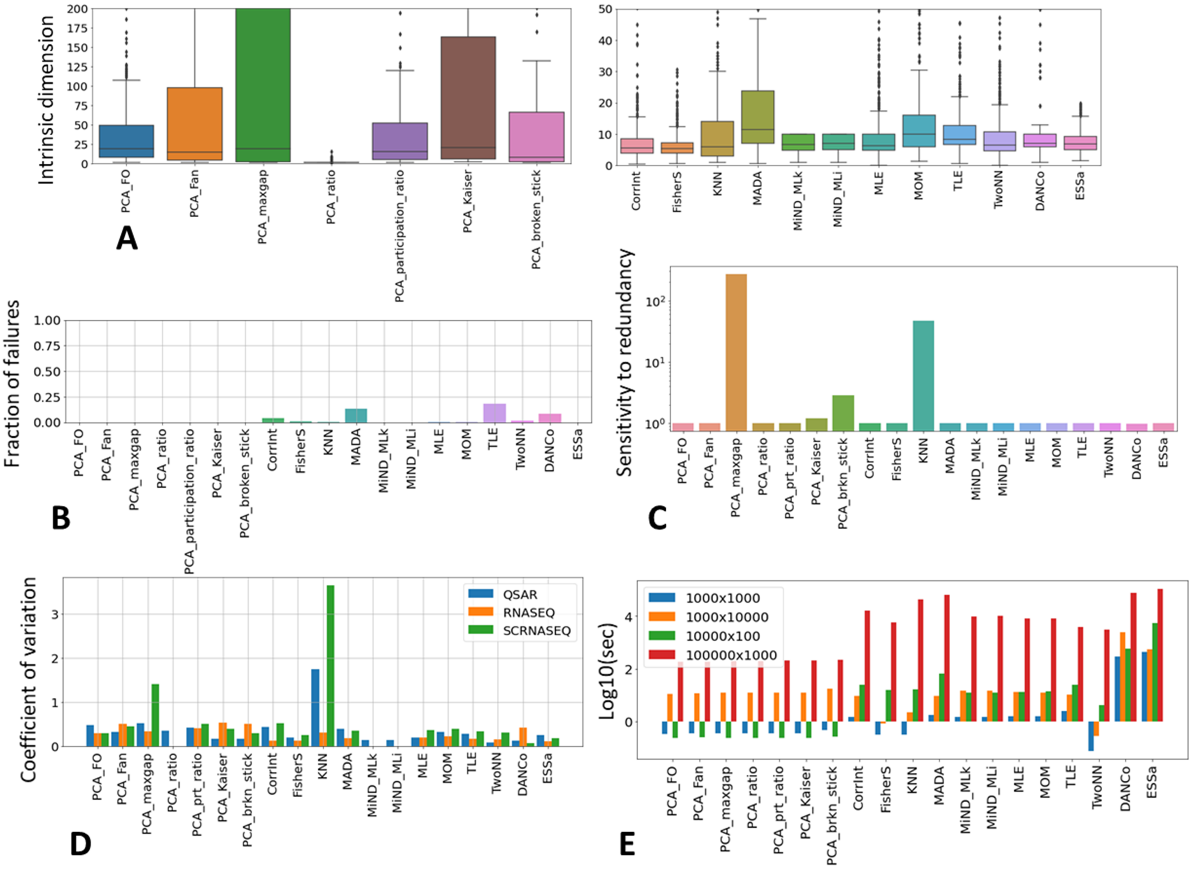


Figure 2. Illustrating different ID method general characteristics: (**A**) range of estimated ID values; (**B**) ability to produce interpretable (positive finite value) result; (**C**) sensitivity to feature redundancy (after duplicating matrix columns); (**D**) uniform ID estimation across datasets of similar nature; (**E**) computational time needed to compute ID for matrices of four characteristic sizes.

3.1.2. Metanalysis of `Scikit-Dimension` ID Estimates

After application of `scikit-dimension`, each dataset was characterized by a vector of 19 measurements of intrinsic dimensionality. The resulting matrix of ID values contained 2.5% missing values, which were imputed, using the standard IterativeImputer from the `sklearn` Python package.

Using the imputed matrix and scaling it to z-scores, we performed principal component analysis (Figure 3A,B). The first principal component explained 42.6% percent of the total variance in ID estimations, with all of the methods having positive and comparable loadings to the first principal component. This justifies the computation of the "consensus" intrinsic dimension measure, which we define here as the mean value of individual ID estimate z-scores. Therefore, the mean ID can take negative or positive values, roughly dividing the datasets into "lower-dimensional" and "higher-dimensional" (Figure 3A,C).

The consensus ID estimate weakly negatively correlated with the number of observations (Pearson $\rho = -0.25$, p-value = 10^{-9}) and positively correlated with the number of features in the dataset (r = 0.44, p-value = 10^{-25}). Nevertheless, even for the datasets with similar matrix shapes, the mean ID estimate could be quite different (Figure 3C).

The second principal component explained 21.3% of the total variance in ID estimates. The loadings of this component roughly differentiated between PCA-based ID estimates and non-linear ID estimation methods, with one exception in the case of the KNN method.

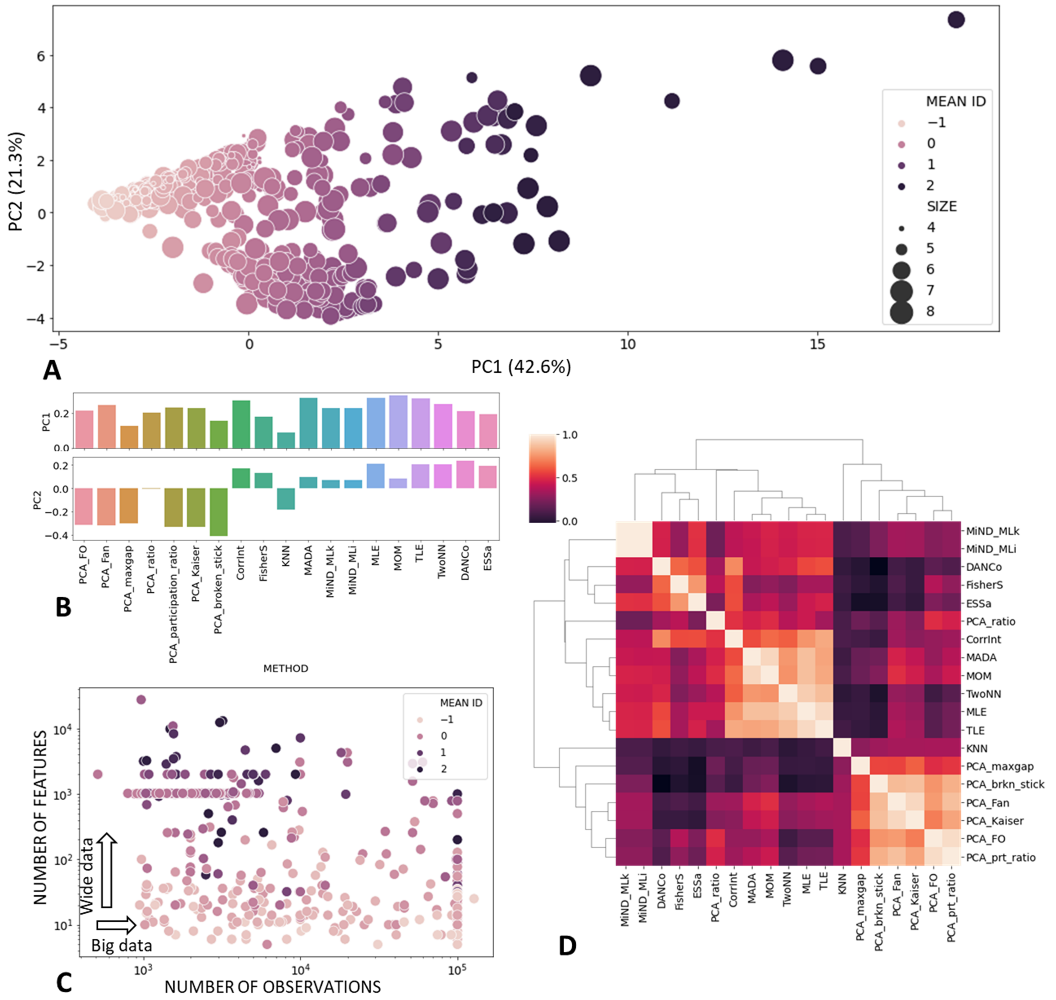


Figure 3. Characterizing `OpenML` dataset collection in terms of ID estimates. (**A**) PCA visualizations of datasets characterized by vectors of 19 ID measures. Size of the point corresponds to the logarithm of the number of matrix entries ($N_{obj} \times N_{var}$). The color corresponds to the mean ID estimate taken as the mean of all ID measure z-scores. (**B**) Loadings of various methods into the first and the second principal component from (**A**). (**C**) Visualization of the mean ID score as a function of data matrix shape. The color is the same as in (**A**). (**D**) Correlation matrix between different ID estimates computed over all analyzed datasets.

We computed the correlation matrix between the results of application of different ID methods (Figure 3D), which also distinguished two large groups of PCA-based and "non-linear" methods. Furthermore, non-linear methods were split into the group of methods, producing results similar to the correlation (fractal) dimension (CorrInt, MADA, MOM, TwoNN, MLE, TLE) and methods based on the concentration of measure phenomena (FisherS, ESS, DANCo, MiND_ML).

In order to illustrate the relation between the dataset geometry and the intrinsic dimension, we produced a gallery of uniform manifold approximation and projection (UMAP) dataset visualizations, with an indication of the ambient dataset dimension (number of features) and the estimated ID, using all methods; Figure 4. One of the conclusions that can be made from this analysis is that the UMAP visualization is not insightful for truly high-dimensional datasets (starting from ID = 10, estimated by the FisherS method). In addition, some datasets, having large ambient dimensions, were characterized with a low ID by most of the methods (e.g., 'hill-valley' dataset).

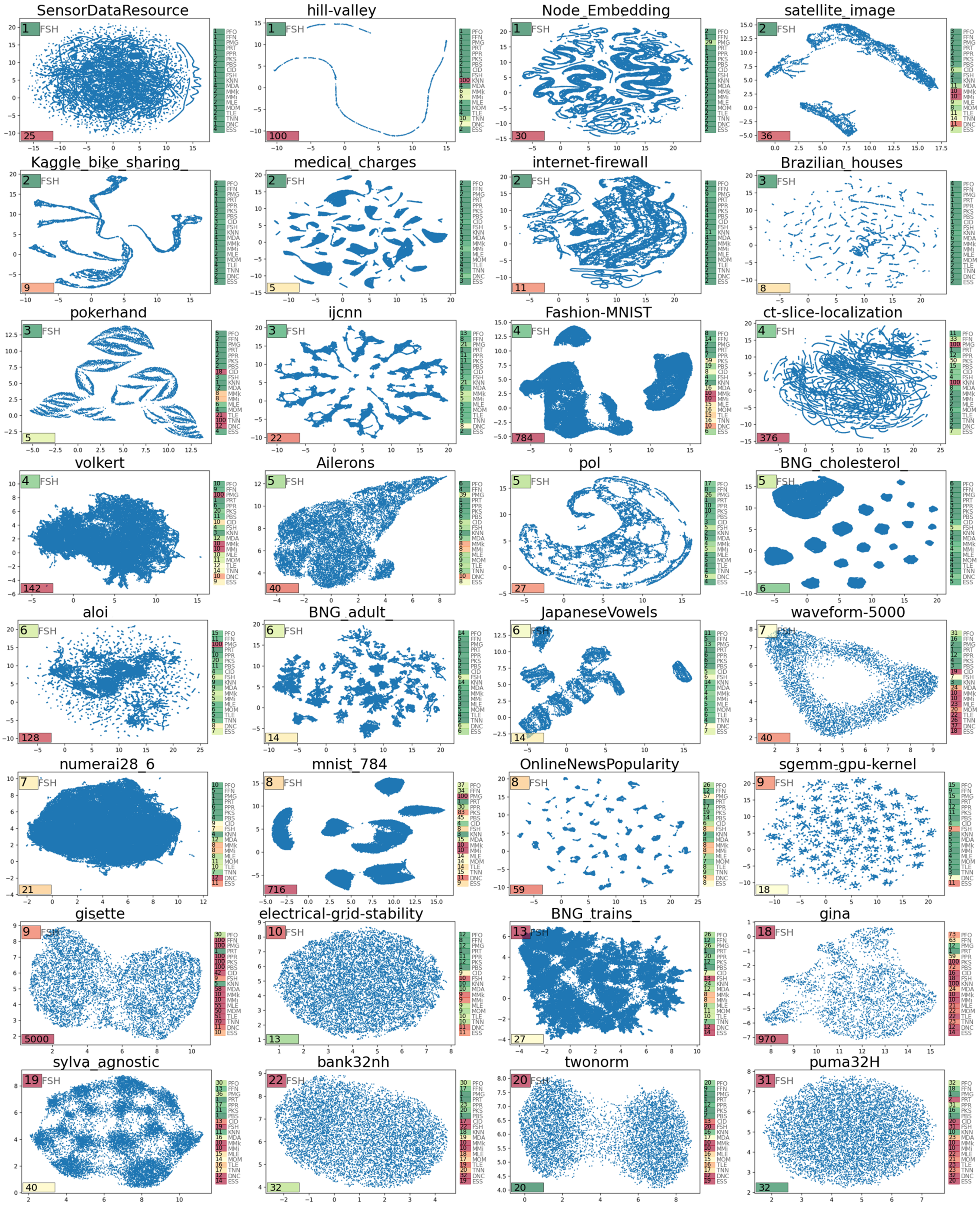


Figure 4. A gallery of UMAP plots computed for a selection of datasets from `OpenML` collection, with indication of ID estimates, ranked by the ID value estimated using Fisher separability-based method (indicated in the left top corner). The ambient dimension of the data (number of features N_{var}) is indicated in the bottom left corner, and the color reflects the ID/N_{var} ratio, from red (close to 0.0 value) to green (close to 1.0). On the right from the UMAP plot, all 19 ID measures are indicated, with color mapped to the value range, from green (small dimension) to red (high dimension).

4. Conclusions

scikit-dimension is to our knowledge the first package implemented in Python, containing implementations of the most-used estimators of ID.

Benchmarking scikit-dimension on a large collection of real-life and synthetic datasets revealed that different estimators of ID possess internal consistency and that the ensemble of ID estimators allows us to achieve more robust classification of datasets into low- or high-dimensionality.

The estimation of intrinsic dimensionality of a dataset is essential in various applications of machine learning to real-life data. We can mention here several typical use cases, where the scikit-dimension package can be used, but this description is by no means comprehensive.

Firstly, learning low-dimensional data geometry (e.g., learning data manifolds or more complex geometries, such as principal graphs [60,61]) frequently requires preliminary data dimensionality reduction for which one has to estimate the 'true' global and local data dimensionality. For example, in the analysis of single-cell data in biology, the inference of so-called cellular trajectories can give different results when more or less principal data dimensions are kept. In higher dimensions, more cell fate decisions can be distinguished, but their inference becomes less robust [62,63]. Some advanced methods of unsupervised learning, such as quantifying the data manifold curvature, require knowledge of data ID [64]. In mathematical modeling of biological and other complex systems, it is frequently important to estimate the effective dimensionality of the dynamical process, from the data or from simulations, in order to inform model reduction [17,65,66]. In medical applications and in the analysis of clinical data, knowledge of consensus data dimensionality was shown to be important to distinguish signal from noise and predict patient trajectories [16].

Secondly, high-dimensional data geometry is a rapidly evolving field in machine learning [67–69]. To know whether the recent theoretical results can be used in practice, one has to estimate the ID of a concrete dataset. More generally, it is important to know if an application of a machine learning method to a dataset will face various types of difficulties, known as the curse of dimensionality. For example, it was shown that, under appropriate assumptions, robustness of general multi-class classifiers to adversarial examples can be achieved only if the intrinsic dimensionality of the AI's decision variables is sufficiently small [70]. Knowledge of ID can be important to decide if one can benefit from the blessing of dimensionality in the problem of correcting the AI's errors when deploying large, pre-trained legacy neural network models [32,71]. Estimating data dimensionality can suggest the application of specific data pre-processing methods, such as hubness reduction of point neighborhood graphs, in the tasks of clustering or non-linear dimensionality reduction [72]. In a recent study, estimating dataset ID was used to show that some old ideas on fighting the curse of dimensionality by modifying global data metrics are not efficient in practice [15]. In this respect, explicit control of the ID of AI models' latent spaces appears to be crucial for developing robust and reliable AI. Our work adds to the spectrum of tools to achieve this aim.

Thirdly, local ID can be used to partition a data point cloud in a way that is complementary to standard clustering [73]. In 3D, this approach can be used for object detection (see Figure 1), but it can be generalized for higher-dimensional data point clouds. Interestingly, local ID can be related to various object characteristics in various domains: folded versus unfolded configurations in a protein molecular dynamics trajectory, active versus non-active regions in brain imaging data, and firms with different financial risk in company balance sheets [74].

Future releases of scikit-dimension will continuously seek to incorporate new estimators and benchmark datasets introduced in the literature, or new features, such as alternative nearest neighbor search for local ID estimates. The package will also include new ID estimators, which can be derived using the most recent achievements in understanding the properties of high-dimensional data geometry [71,75].

Author Contributions: Conceptualization, J.B., A.Z., I.T., A.N.G.; methodology, J.B., E.M.M., A.N.G.; software, J.B., E.M.M. and A.Z.; formal analysis, J.B., E.M.M., A.Z.; data curation, J.B. and A.Z.; writing—original draft preparation, J.B. and A.Z.; writing—review and editing, all authors; supervision, A.Z. and A.N.G. All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported by the Ministry of Science and Higher Education of the Russian Federation (Project No. 075-15-2020-927), by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'Avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute), by the Association Science et Technologie, the Institut de Recherches Internationales Servier and the doctoral school Frontières de l'Innovation en Recherche et Education Programme Bettencourt. I.T. was supported by the UKRI Turing AI Acceleration Fellowship (EP/V025295/1).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this study were retrieved from public sources, namely the `OpenML` repository, CytoTRACE website https://cytotrace.stanford.edu/ (section "Downloads", accessed on 18 October 2021), from the Data Portal of National Cancer Institute https://portal.gdc.cancer.gov/ (accessed on 18 October 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
2. Bishop, C.M. *Neural Networks for Pattern Recognition*; Oxford University Press: Oxford, UK, 1995. [CrossRef]
3. Fukunaga, K. Intrinsic dimensionality extraction. In *Pattern Recognition and Reduction of Dimensionality, Handbook of Statistics*; Krishnaiah, P.R., Kanal, L.N., Eds.; North-Holland: Amsterdam, The Netherlands, 1982; Volume 2, pp. 347–362.
4. Albergante, L.; Bac, J.; Zinovyev, A. Estimating the effective dimension of large biological datasets using Fisher separability analysis. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
5. Giudice, M.D. Effective Dimensionality: A Tutorial. *Multivar. Behav. Res.* **2020**, 1–16. [CrossRef]
6. Palla, K.; Knowles, D.; Ghahramani, Z. A nonparametric variable clustering model. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2012; Volume 4, pp. 2987–2995.
7. Giuliani, A.; Benigni, R.; Sirabella, P.; Zbilut, J.P.; Colosimo, A. Nonlinear Methods in the Analysis of Protein Sequences: A Case Study in Rubredoxins. *Biophys. J.* **2000**, *78*, 136–149. [CrossRef]
8. Jiang, H.; Kim, B.; Guan, M.Y.; Gupta, M.R. To Trust Or Not To Trust A Classifier. In *NeurIPS*; Montreal Convention Centre: Montreal, QC, Canada, 2018; pp. 5546–5557. [CrossRef]
9. Bac, J.; Zinovyev, A. Lizard Brain: Tackling Locally Low-Dimensional Yet Globally Complex Organization of Multi-Dimensional Datasets. *Front. Neurorobotics* **2020**, *13*, 110. [CrossRef]
10. Hino, H. ider: Intrinsic Dimension Estimation with R. *R J.* **2017**, *9*, 329–341. [CrossRef]
11. Campadelli, P.; Casiraghi, E.; Ceruti, C.; Rozza, A. Intrinsic Dimension Estimation: Relevant Techniques and a Benchmark Framework. *Math. Probl. Eng.* **2015**, *2015*, 759567. [CrossRef]
12. Camastra, F.; Staiano, A. Intrinsic dimension estimation: Advances and open problems. *Inf. Sci.* **2016**, *328*, 26–41. [CrossRef]
13. Little, A.V.; Lee, J.; Jung, Y.; Maggioni, M. Estimation of intrinsic dimensionality of samples from noisy low-dimensional manifolds in high dimensions with multiscale SVD. In Proceedings of the 2009 IEEE/SP 15th Workshop on Statistical Signal Processing, Cardiff, UK, 31 August–3 September 2009; pp. 85–88. [CrossRef]
14. Hein, M.; Audibert, J.Y. Intrinsic dimensionality estimation of submanifolds in R^d. In Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, 7–11 August 2005; ACM: New York, NY, USA, 2005; pp. 289–296. [CrossRef]
15. Mirkes, E.; Allohibi, J.; Gorban, A.N. Fractional Norms and Quasinorms Do Not Help to Overcome the Curse of Dimensionality. *Entropy* **2020**, *22*, 1105. [CrossRef] [PubMed]
16. Golovenkin, S.E.; Bac, J.; Chervov, A.; Mirkes, E.M.; Orlova, Y.V.; Barillot, E.; Gorban, A.N.; Zinovyev, A. Trajectories, bifurcations, and pseudo-time in large clinical datasets: Applications to myocardial infarction and diabetes data. *GigaScience* **2020**, *9*, giaa128. [CrossRef] [PubMed]
17. Zinovyev, A.; Sadovsky, M.; Calzone, L.; Fouché, A.; Groeneveld, C.S.; Chervov, A.; Barillot, E.; Gorban, A.N. Modeling Progression of Single Cell Populations Through the Cell Cycle as a Sequence of Switches. *bioRxiv* **2021**. [CrossRef]
18. Grassberger, P.; Procaccia, I. Measuring the strangeness of strange attractors. *Phys. D Nonlinear Phenom.* **1983**, *9*, 189–208. [CrossRef]

19. Farahmand, A.M.; Szepesvári, C.; Audibert, J.Y. Manifold-adaptive dimension estimation. In Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, USA, 20–24 June 2007; pp. 265–272. [CrossRef]
20. Amsaleg, L.; Chelly, O.; Furon, T.; Girard, S.; Houle, M.E.; Kawarabayashi, K.; Nett, M. Extreme-value-theoretic estimation of local intrinsic dimensionality. *Data Min. Knowl. Discov.* **2018**, *32*, 1768–1805. [CrossRef]
21. Jackson, D.A. Stopping rules in principal components analysis: A comparison of heuristical and statistical approaches. *Ecology* **1993**, *74*, 2204–2214. [CrossRef]
22. Fukunaga, K.; Olsen, D.R. An Algorithm for Finding Intrinsic Dimensionality of Data. *IEEE Trans. Comput.* **1971**, *C-20*, 176–183. [CrossRef]
23. Mingyu, F.; Gu, N.; Qiao, H.; Zhang, B. Intrinsic dimension estimation of data by principal component analysis. *arXiv* **2010**, arXiv:1002.2050.
24. Hill, B.M. A simple general approach to inference about the tail of a distribution. *Ann. Stat.* **1975**, 1163–1174. [CrossRef]
25. Levina, E.; Bickel, P.J. Maximum Likelihood estimation of intrinsic dimension. In Proceedings of the 17th International Conference on Neural Information Processing Systems (Vancouver, Canada, 1 December 2004); MIT Press: Cambridge, MA, USA, 2004; pp. 777–784. [CrossRef]
26. Haro, G.; Randall, G.; Sapiro, G. Translated poisson mixture model for stratification learning. *Int. J. Comput. Vis.* **2008**, *80*, 358–374. [CrossRef]
27. Carter, K.M.; Raich, R.; Hero, A.O. On Local Intrinsic Dimension Estimation and Its Applications. *IEEE Trans. Signal Process.* **2010**, *58*, 650–663. [CrossRef]
28. Rozza, A.; Lombardi, G.; Ceruti, C.; Casiraghi, E.; Campadelli, P. Novel high intrinsic dimensionality estimators. *Mach. Learn.* **2012**, *89*, 37–65. [CrossRef]
29. Ceruti, C.; Bassis, S.; Rozza, A.; Lombardi, G.; Casiraghi, E.; Campadelli, P. DANCo: An intrinsic dimensionality estimator exploiting angle and norm concentration. *Pattern Recognit.* **2014**, *47*, 2569–2581. [CrossRef]
30. Johnsson, K. Structures in High-Dimensional Data: Intrinsic Dimension and Cluster Analysis. Ph.D. Thesis, Faculty of Engineering, LTH, Perth, Australia, 2016.
31. Facco, E.; D'Errico, M.; Rodriguez, A.; Laio, A. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Sci. Rep.* **2017**, *7*, 12140. [CrossRef] [PubMed]
32. Gorban, A.; Golubkov, A.; Grechuk, B.; Mirkes, E.; Tyukin, I. Correction of AI systems by linear discriminants: Probabilistic foundations. *Inf. Sci.* **2018**, *466*, 303–322. [CrossRef]
33. Amsaleg, L.; Chelly, O.; Houle, M.E.; Kawarabayashi, K.; Radovanović, M.; Treeratanajaru, W. Intrinsic dimensionality estimation within tight localities. In Proceedings of the 2019 SIAM International Conference on Data Mining, Calgary, AB, Canada, 2–4 May 2019; SIAM: Philadelphia, PA, USA, 2019; pp. 181–189.
34. Harris, C.R.; Millman, K.J.; van der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. [CrossRef]
35. Hunter, J.D. Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.* **2007**, *9*, 90–95. [CrossRef]
36. The Pandas Development Team. Pandas-Dev/Pandas: Pandas 1.3.4, Zenodo. Available online: https://zenodo.org/record/5574486#.YW50jhpByUk (accessed on 18 October 2021). [CrossRef]
37. Lam, S.K.; Pitrou, A.; Seibert, S. Numba: A llvm-based python jit compiler. In Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, Austin, TX, USA, 15 November 2015; pp. 1–6. [CrossRef]
38. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **2020**, *17*, 261–272. [CrossRef] [PubMed]
39. Johnsson, K. intrinsicDimension: Intrinsic Dimension Estimation (R Package). 2019. Available online https://rdrr.io/cran/intrinsicDimension/ (accessed on 6 September 2021).
40. You, K. Rdimtools: An R package for Dimension Reduction and Intrinsic Dimension Estimation. *arXiv* **2020**, arXiv:2005.11107.
41. Denti, Francesco intRinsic: An R package for model-based estimation of the intrinsic dimension of a dataset. *arXiv* **2021**, arXiv:2102.11425.
42. Hein, M.J.Y.A. IntDim: Intrindic Dimensionality Estimation. 2016. Available online: https://www.ml.uni-saarland.de/code/IntDim/IntDim.htm (accessed on 6 September 2021).
43. Lombardi, G. Intrinsic Dimensionality Estimation Techniques (MATLAB Package). 2013. Available online: https://fr.mathworks.com/matlabcentral/fileexchange/40112-intrinsic-dimensionality-estimation-techniques (accessed on 6 September 2021).
44. van der Maaten, L. Drtoolbox: Matlab Toolbox for Dimensionality Reduction. 2020. Available online: https://lvdmaaten.github.io/drtoolbox/ (accessed on 6 September 2021).
45. Radovanović, M. Tight Local Intrinsic Dimensionality Estimator (TLE) (MATLAB Package). 2020. Available online: https://perun.pmf.uns.ac.rs/radovanovic/tle/ (accessed on 6 September 2021).
46. Gomtsyan, M.; Mokrov, N.; Panov M.; Yanovich Y. Geometry-Aware Maximum Likelihood Estimation of Intrinsic Dimension (Python Package). 2019. Available online: https://github.com/stat-ml/GeoMLE (accessed on 6 September 2021).
47. Gomtsyan, M.; Mokrov, N.; Panov, M.; Yanovich, Y. Geometry-Aware Maximum Likelihood Estimation of Intrinsic Dimension. 2019. In Proceedings of the Eleventh Asian Conference on Machine Learning, Nagoya, Japan, 17–19 November 2019; pp. 1126–1141.

48. Erba, V. pyFCI: A Package for Multiscale-Full-Correlation-Integral Intrinsic Dimension Estimation. 2019. Available online: https://github.com/vittorioerba/pyFCI (accessed on 6 September 2021).
49. Granata, D. Intrinsic-Dimension (Python Package). 2016. Available online: https://github.com/dgranata/Intrinsic-Dimension (accessed on 6 September 2021).
50. Bac, J.; Zinovyev, A. Local intrinsic dimensionality estimators based on concentration of measure. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8.
51. Gorban, A.N.; Makarov, V.A.; Tyukin, I.Y. The unreasonable effectiveness of small neural ensembles in high-dimensional brain. *Phys. Life Rev.* **2019**, *29*, 55–88. [CrossRef]
52. Vanschoren, J.; van Rijn, J.N.; Bischl, B.; Torgo, L. OpenML: Networked Science in Machine Learning. *SIGKDD Explor.* **2013**, *15*, 49–60. [CrossRef]
53. Gulati, G.; Sikandar, S.; Wesche, D.; Manjunath, A.; Bharadwaj, A.; Berger, M.; Ilagan, F.; Kuo, A.; Hsieh, R.; Cai, S.; et al. Single-cell transcriptional diversity is a hallmark of developmental potential. *Science* **2020**, *24*, 405–411. [CrossRef]
54. Giuliani, A. The application of principal component analysis to drug discovery and biomedical data. *Drug Discov. Today* **2017**, *22*, 1069–1076. [CrossRef] [PubMed]
55. Cangelosi, R.; Goriely, A. Component retention in principal component analysis with application to cDNA microarray data. *Biol. Direct* **2007**, *2*, 2. [CrossRef]
56. Johnsson, K.; Soneson, C.; Fontes, M. Low Bias Local Intrinsic Dimension Estimation from Expected Simplex Skewness. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 196–202. [CrossRef] [PubMed]
57. Jolliffe, I.T. *Principal Component Analysis*; Springer Series in Statistics; Springer: Berlin/Heidelberg, Germany, 2002.
58. Kaiser, H. The Application of Electronic Computers to Factor Analysis. *Educ. Psychol. Meas.* **1960**, *20*, 141 – 151. [CrossRef]
59. Frontier, S. Étude de la décroissance des valeurs propres dans une analyse en composantes principales: Comparaison avec le modèle du bâton brisé. *J. Exp. Mar. Biol. Ecol.* **1976**, *25*, 67–75. [CrossRef]
60. Gorban, A.N.; Sumner, N.R.; Zinovyev, A.Y. Topological grammars for data approximation. *Appl. Math. Lett.* **2007**, *20*, 382–386. [CrossRef]
61. Albergante, L.; Mirkes, E.; Bac, J.; Chen, H.; Martin, A.; Faure, L.; Barillot, E.; Pinello, L.; Gorban, A.; Zinovyev, A. Robust and scalable learning of complex intrinsic dataset geometry via ElPiGraph. *Entropy* **2020**, *22*, 296. [CrossRef]
62. Lähnemann, D.; Köster, J.; Szczurek, E.; McCarthy, D.J.; Hicks, S.C.; Robinson, M.D.; Vallejos, C.A.; Campbell, K.R.; Beerenwinkel, N.; Mahfouz, A.; et al. Eleven grand challenges in single-cell data science. *Genome Biol.* **2020**, *21*, 1–31. [CrossRef]
63. Chen, H.; Albergante, L.; Hsu, J.Y.; Lareau, C.A.; Lo Bosco, G.; Guan, J.; Zhou, S.; Gorban, A.N.; Bauer, D.E.; Aryee, M.J.; et al. Single-cell trajectories reconstruction, exploration and mapping of omics data with STREAM. *Nat. Commun.* **2019**, *10*, 1–14. [CrossRef]
64. Sritharan, D.; Wang, S.; Hormoz, S. Computing the Riemannian curvature of image patch and single-cell RNA sequencing data manifolds using extrinsic differential geometry. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2100473118. [CrossRef]
65. Radulescu, O.; Gorban, A.N.; Zinovyev, A.; Lilienbaum, A. Robust simplifications of multiscale biochemical networks. *BMC Syst. Biol.* **2008**, *2*, 86. [CrossRef]
66. Gorban, A.N.; Zinovyev, A. Principal manifolds and graphs in practice: From molecular biology to dynamical systems. *Int. J. Neural Syst.* **2010**, *20*, 219–232. [CrossRef]
67. Donoho, D.L. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Math Challenges Lect.* **2000**, *1*, 1–32.
68. Gorban, A.N.; Tyukin, I.Y. Blessing of dimensionality: Mathematical foundations of the statistical physics of data. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2018**, *376*, 20170237. [CrossRef]
69. Kainen, P.C.; Kůrková, V. Quasiorthogonal dimension of euclidean spaces. *Appl. Math. Lett.* **1993**, *6*, 7–10. [CrossRef]
70. Tyukin, I.Y.; Higham, D.J.; Gorban, A.N. On Adversarial Examples and Stealth Attacks in Artificial Intelligence Systems. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–6.
71. Gorban, A.N.; Grechuk, B.; Mirkes, E.M.; Stasenko, S.V.; Tyukin, I.Y. High-Dimensional Separability for One- and Few-Shot Learning. *Entropy* **2021**, *23*, 1090. [CrossRef] [PubMed]
72. Amblard, E.; Bac, J.; Chervov, A.; Soumelis, V.; Zinovyev, A. Hubness reduction improves clustering and trajectory inference in single-cell transcriptomic data. *bioRxiv* **2021**. [CrossRef]
73. Gionis, A.; Hinneburg, A.; Papadimitriou, S.; Tsaparas, P. Dimension Induced Clustering. In *KDD '05: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*; Association for Computing Machinery: New York, NY, USA, 2005; pp. 51–60. [CrossRef]
74. Allegra, M.; Facco, E.; Denti, F.; Laio, A.; Mira, A. Data segmentation based on the local intrinsic dimension. *Sci. Rep.* **2020**, *10*, 1–12. [CrossRef] [PubMed]
75. Grechuk, B.; Gorban, A.N.; Tyukin, I.Y. General stochastic separation theorems with optimal bounds. *Neural Netw.* **2021**, *138*, 33–56. [CrossRef]