Journal of Machine Learning Research X (2016) X-XX Submitted X/XX; Published XX/XX

# ADENINE — A Data ExploratioN pIpeliNE

**Samuele Fiorini** samuele.fiorini@dibris.unige.it
**Federico Tomasi** federico.tomasi@dibris.unige.it
**Annalisa Barla** annalisa.barla@unige.it

*Department of Informatics, Bioengineering,*
*Robotics and System Engineering (DIBRIS)*
*University of Genoa*
*Genoa, I-16146, Italy*

**Editor:** Editor name

## Abstract

In this paper we introduce `Adenine`, a `Python` framework for data exploration. The main goal of `Adenine`, is twofold: helping researchers and data scientists achieving a first and quick overview on the main structures underlying their data and choosing the most suitable unsupervised learning pipeline for the problem at hand. This software tool encompasses state of the art techniques for: missing values imputing, preprocessing, dimensionality reduction and clustering tasks. `Adenine` exploits both process and thread level parallelism and it is capable of generating nice and clean publication-ready figures along with a quantitative description of the algorithms performance. `Adenine` is released under FreeBSD license and it can be downloaded from http://slipguru.github.io/adenine/.

**Keywords:** Data exploration, unsupervised learning, RNA-Seq gene expression

## 1. Introduction

## 2. Implementation

From an implemetative standpoint, `Adenine` is built around the concept of *pipeline*, that is a sequence of four fundamental steps: (i) missing values imputing, (ii) data preprocessing, (iii) dimensionality reduction and (iv) clustering (see Figure 1).

For each step, a fair number of off-the-shelf algorithms are available (see Table 2). The vast majority of their implementations is inherited, or extended, from `scikit-learn` (Pedregosa et al., 2011).

In order to perform exploratory analysis of large datasets, `Adenine` can take advantage of different parallel computing paradigms. For instance, its pipelines are designed to be independent from each other, therefore they all run in parallel as separate `Python` processes on different cores. Moreover, since `Adenine` makes large use of `numpy` and `scipy`, it automatically benefits from their bindings with optimized linear algebra libraries (such as OpenBLAS[1], or Intel® MKL).

1. http://www.openblas.net/

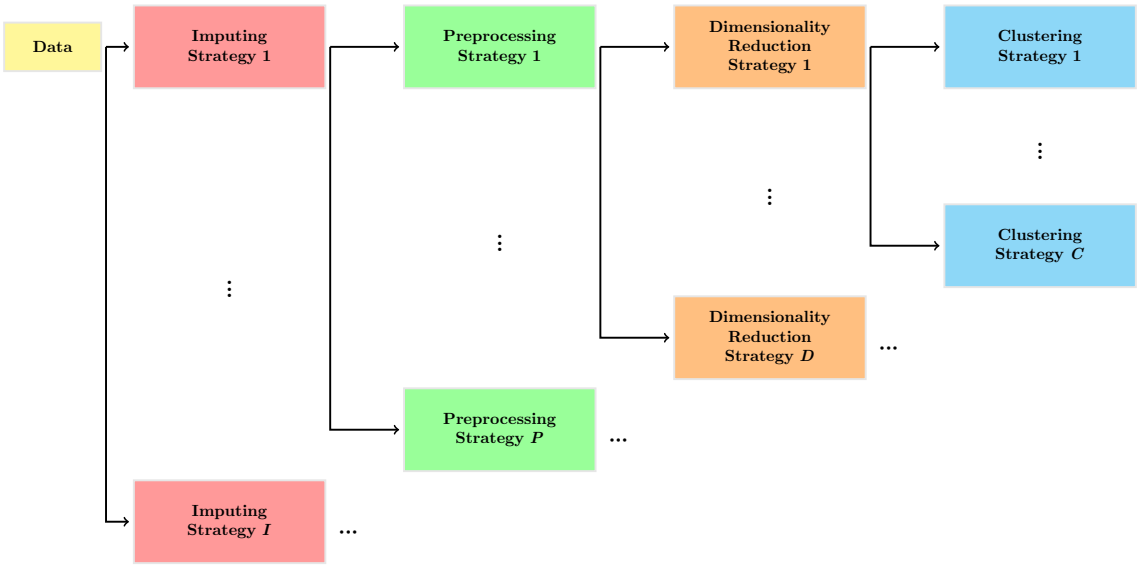


Figure 1: A schematic representation of the `Adenine` workflow. The list of building blocks available for each step is summarized in Table 1.

## 3. Experiments and results

To assess the quality of the obtained results, we tested `Adenine` on a set of synthetic and real dataset.

**{parla qui dei test synth} {TGCA}**

## 4. Conclusions

## References

Christopher M Bishop. Pattern recognition. *Machine Learning*, 2006.

Ingwer Borg and Patrick JF Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

Table 1: Pipelines building blocks and relative references (which are not reported when the definition is given in Section 2).

| **Step** | **Algorithms** | **Ref.** |
|---|---|---|
| Imputing | Mean | |
| | Median | |
| | KNN | (Troyanskaya et al., 2001) |
| Preprocessing | Recentering | |
| | Standardize | |
| | Normalize | |
| | MinMax | |
| Dimensionality reduction | Principal component Analysis (PCA) | (Jolliffe, 2002) |
| | Incremental PCA | (Ross et al., 2008) |
| | Randomized PCA | (Halko et al., 2011) |
| | Kernel PCA | (Schölkopf et al., 1997) |
| | Isomap | (Tenenbaum et al., 2000) |
| | Locally linear embedding | (Roweis and Saul, 2000) |
| | Spectral embedding | (Ng et al., 2002) |
| | Multidimensional scaling | (Borg and Groenen, 2005) |
| | t-Distributed Stochastic Neighbor Embedding (t-SNE) | (Van der Maaten and Hinton, 2008) |
| Clustering | K-means | (Bishop, 2006) |
| | Affinity propagation | (Frey and Dueck, 2007) |
| | Mean Shift | (Comaniciu and Meer, 2002) |
| | Spectral | (Shi and Malik, 2000) |
| | Hierarchical | (Friedman et al., 2001) |

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008.

Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *Artificial Neural NetworksICANN'97*, pages 583–588. Springer, 1997.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.