# Supplementary Material

Ruriko Yoshida Kenji Fukumizu Chrysafis Vogiatzis

February 24, 2016

## 1 Clustering and dimension reduction methods

### 1.1 Normalized cut

The input to the normalized cut framework is a weighted graph, with each edge weight expressing the dissimilarity between the nodes. More precisely, let $G = (V, E)$ be a finite undirected weighted graph, where $V = \{1, \ldots, n\}$ is the node set and $E$ is the edge set. Each node $i$ corresponds to a data $X_i$, and the weight $w_{ij}$ of an edge $(i, j) \in E$ represents the similarity of $X_i$ and $X_j$. The goal of the normalized cuts is to provide a partition of $V$ into $K$ sets $V_1, \ldots, V_K$ ($\cup_{a=1}^{K} V_a = V$ and $V_a \cap V_b = \emptyset$ for $a \neq b$) such that all the nodes in $V_a$ are similar while the nodes in $V_a$ and the nodes in $V_b$ are dissimilar.

For simplicity, consider clustering of the nodes into two clusters $A$ and $B$. To measure the total goodness of this partition, the normalized cut is defined as a normalization of Cut criterion $\mathrm{Cut}(A, B)$, which is defined by

$$\mathrm{Cut}(A, B) := \sum_{i \in A, j \in B} w_{ij}.$$

In graph theory, the problem of partitioning the nodes to minimize $\mathrm{Cut}(A, B)$ is called *minimum cut*. It is a well-studied problem, and efficient algorithms are known.

While the minimum cut can be a criterion of clustering, it is known that it tends to make a small cluster of isolated nodes. To solve this problem, Shi and Malik (2000) introduces the normalized cut $\mathrm{Ncut}(A, B)$ defined by

$$\mathrm{Ncut}(A, B) = \frac{\mathrm{Cut}(A, B)}{\mathrm{assoc}(A, V)} + \frac{\mathrm{Cut}(A, B)}{\mathrm{assoc}(B, V)},$$

where $\mathrm{assoc}(A, V) = \sum_{i \in A, j \in V} w_{ij}$ and $\mathrm{assoc}(B, V)$ is, of course, computed accordingly. The normalization by assoc avoids the large value of Ncut for a cluster of a small number of isolated points.

Computation of minimum Ncut is NP-complete if rigorously implemented, but a reasonable approximation with eigendecomposition is available. Let $W$ be a symmetric matrix with elements $w_{ij}$, and $D$ be a diagonal matrix with $D_{ii} = \sum_{j=1}^{n} W_{ij}$. Consider for simplicity the case of two clusters. By introducing a vector $u \in \{\pm 1\}^n$, where $u_i = 1$ and $u_i = -1$ mean $i \in A$ and $i \in B$, respectively, the Ncut can be restated by

$$\mathrm{Ncut}(A, B) = \frac{-\sum_{u_i > 0, u_j < 0} W_{ij} u_i u_j}{\sum_{u_i > 0} D_{ii}} + \frac{-\sum_{u_i > 0, u_j < 0} W_{ij} u_i u_j}{\sum_{u_i < 0} D_{ii}}.$$

Introducing a binary vector $y$ such that $y_i = 1$ if $u_i = 1$ and $y_i = -b$ if $u_i = -1$, where $b = \sum_{u_i > 0} D_{ii} / \sum_{u_j < 0} D_{jj}$, one can see that the minimum Ncut is equivalent to

$$\min_{y \in \{1, -b\}^n} \frac{y^T (D - W) y}{y^T D y}$$

under the constraint

$$y^T D \mathbf{1}_n = 0$$

with $\mathbf{1}_n = (1, \ldots, 1)^T$. If we relax the condition $y \in \{1, -b\}^n$ by allowing continuous values for $y$, the above minimization is solved by the generalized eigenproblem:

$$(D - W) y = \lambda D y.$$

Note that this problem has a trivial solution $y = \mathbf{1}_n$ and $\lambda = 0$. The constraint $y^T D \mathbf{1}_n = 0$ thus implies that the relaxed problem is solved by the eigenvector corresponding to the second smallest eigenvalue. An approximated solution to the normalized cut is then given by the discretization of the continuous $y$.

When we consider more than two clusters, we can apply the procedure recursively to the segmented parts. It is known that this is equivalent to use the eigenvectors corresponding to the other smallest eigenvalues for partitioning. For the details of the algorithm, see Shi and Malik (2000).

## 1.2 Kernel PCA

Kernel PCA (Schölkopf *et al.*, 1998) is one of the kernel methods, which use a positive definite kernel to define a feature map for extracting nonlinearity of data. A positive definite kernel $k(x, y)$ is a function with two

arguments such that the Gram matrix $(k(x_i, x_j))_{i,j=1}^n$ is positive semidefinite for any points $x_1, \ldots, x_n$ in the data space. A kernel method in general applies some method of data analysis such as PCA to the feature vectors $\phi(X_i), \ldots, \phi(X_N)$, where $\phi$ is a feature map from the data space to a feature space. The feature space $\mathcal{H}$ is a function space, called reproducing kernel Hilbert space, determined uniquely by the kernel $k$. It is known that the Hilbert space $\mathcal{H}$ has a special inner product that satisfies the reproducing property; $\langle f, k(\cdot, x)\rangle = f(x)$ for any $f \in \mathcal{H}$ and point $x$. The feature map $\phi$ is defined by $\phi(x) = k(\cdot, x) \in \mathcal{H}$ so that the inner product of two feature vectors can be evaluated simply by the kernel value, i.e., $\langle \phi(x), \phi(y)\rangle = k(x, y)$.

With the simplified computation of the inner product, it is known (Schölkopf *et al.*, 1998) that the PCA on the feature vectors can be solved by the following generalized eigenproblem:

$$K^2 u = \lambda K u \qquad \text{subject to} \quad u^T K u = 1 \tag{1}$$

where $K$ is the centered Gram matrix define by

$$K_{ij} = k(X_i, X_j) - \frac{1}{n}\sum_{b=1}^{n} k(X_i, X_b) - -\frac{1}{n}\sum_{a=1}^{n} k(X_a, X_j) + \frac{1}{n^2}\sum_{a,b=1}^{n} k(X_a, X_b).$$

The $p$-th principal component of data point $X_i$ is then given by

$$\sqrt{\lambda_p} u_{ip},$$

where $u_p$ is the unit eigenvector of $K$ corresponding to the $p$-th largest eigenvalue $\lambda_p$.

Kernel PCA has been applied to a wide variety of problems for the purpose of dimension reduction and data visualization. To list a few, they include a biological applications such as Popescu *et al.* (2014) and Reverter *et al.* (2014).

### 1.3 t-SNE

The t-SNE is a method of dimension reduction especially for data visualization (van der Maaten and Hinton, 2008). Given data points $X_1, \ldots, X_n$ in a high dimensional Euclidean space, t-SNE first computes the probability $p_{j|i}$ that $X_j$ is a neighbor of $X_i$:

$$p_{j|i} = \frac{\exp(-\frac{1}{2\sigma_i^2}\|X_i - X_j\|^2)}{\sum_{k\neq i}\exp(-\frac{1}{2\sigma_i^2}\|X_i - X_k\|^2)} \qquad (j \neq i)$$

and $p_{i|i} = 0$, where $\sigma_i^2$ is a bandwidth parameter depending on $i$. The similarity between $X_i$ and $X_j$ are then measured by

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}.$$

The goal of t-SNE is to provide a set of low dimensional representation $Y_1, \ldots, Y_n$ such that their similarities are close to $p_{ij}$. The similarity of $Y_i$ and $Y_j$ are defined by

$$q_{ij} = \frac{(1 + \|Y_i - Y_j\|^2)^{-1}}{\sum_{k \neq \ell}(1 + \|Y_k - Y_\ell\|^2)^{-1}}.$$

Unlike $p_{j|i}$, the above similarity $q_{ij}$ uses the form of heavy-tailed $t$-distribution. This aims at avoiding a so-called *crowding problem*: since a larger number of points can be equally similar in a higher-dimensional space, it is impossible to locate all such points in a lower-dimensional space. The heavy-tailed $q_{ij}$ can assign larger similarity for apart points than the Gaussian-type similarity so that slightly more dissimilar objects can be located far apart. This works as a solution to the crowding problem.

The locations $(Y_i)$ are optimized so that the Kullback-Leibler divergence of $(q_{ij})$ from $(p_{ij})$ is minimized;

$$\min_{(Y_i)} \sum_{ij} p_{ij} \log \frac{p_{ij}}{q_{ij}}.$$

A gradient-based method is applied to numerical optimization of this objective function. A method for accelerating the optimization has been also proposed (van der Maaten, 2014).

The t-SNE method is known to work effectively for visualization of data in a two or three dimensional space, and has been applied to various problems including biological ones (Platzer, 2013, Amir *et al.*, 2013, Saadatpour *et al.*, 2014).

## 2 Comparison of clustering accuracy

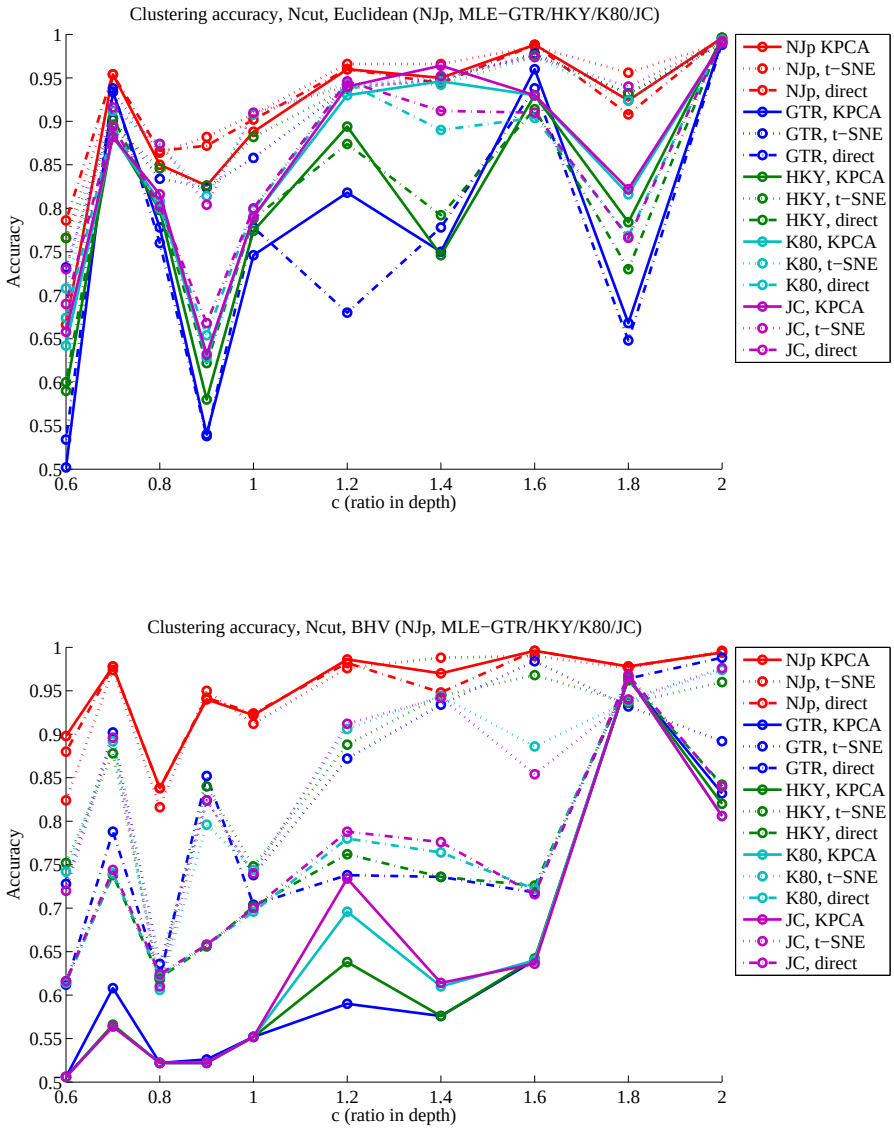


Figure 1: Comparison of Ncut clustering accuracy between Euclidean distance and BHV tree space distance.

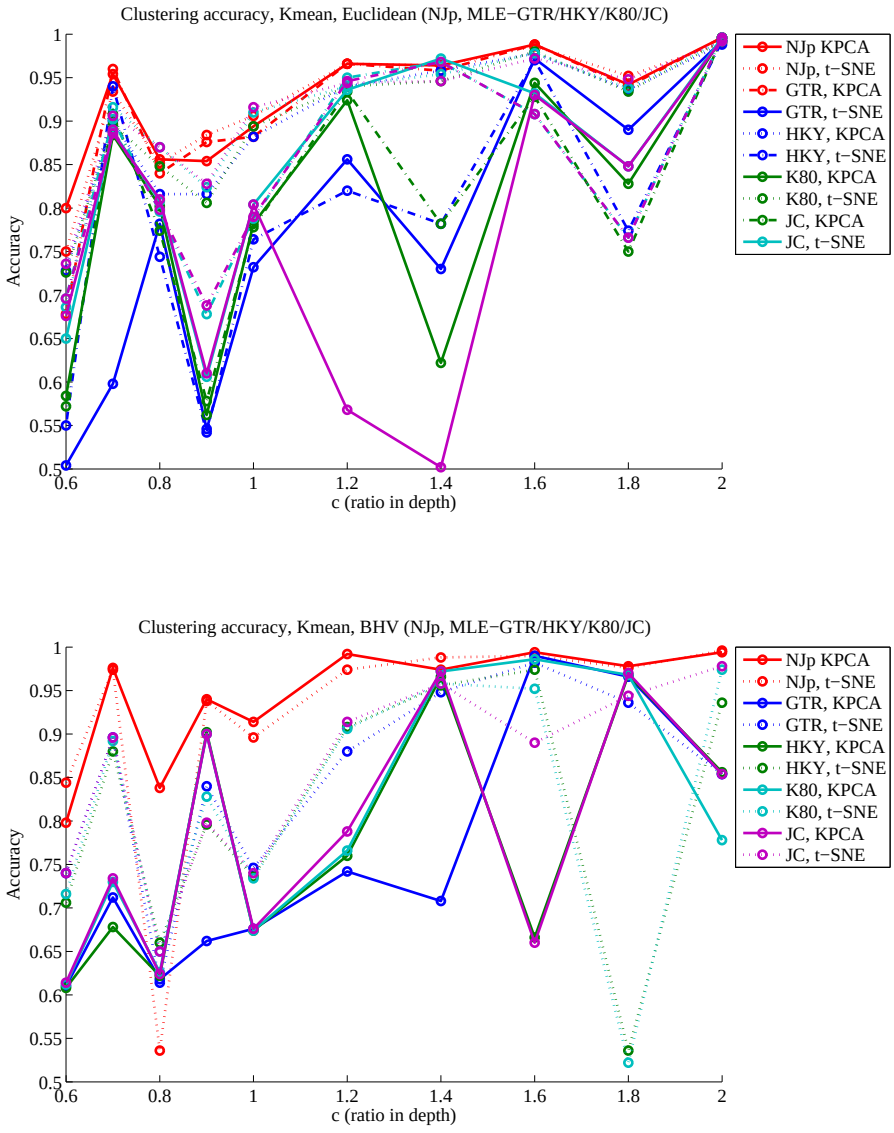


Figure 2: Comparison of K means clustering accuracy between Euclidean distance and BHV tree space distance.

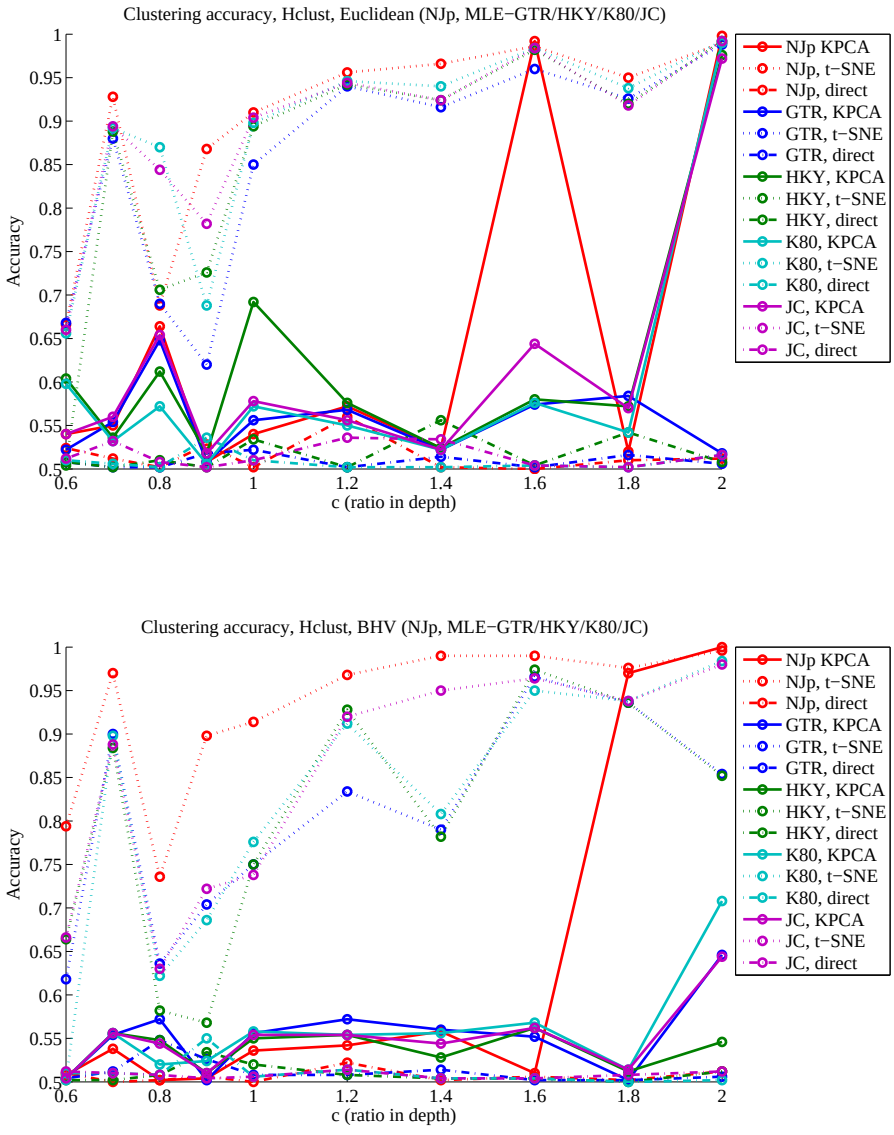


Figure 3: Comparison of hierarchical clustering accuracy between Euclidean distance and BHV tree space distance.

| Method | Dim. Red. | Distance | $c$ 0.6 | 0.8 | 1.2 |
|---|---|---|---|---|---|
| NJp | KPCA | BHV | 0.898 | 0.838 | 0.986 |
| | | Euclid | 0.666 | 0.850 | 0.960 |
| | t-SNE | BHV | 0.848 | 0.822 | 0.970 |
| | | Euclid | 0.732 | 0.860 | 0.962 |
| | Isomap | BHV | 0.722 | 0.570 | 0.976 |
| | | Euclid | 0.790 | 0.870 | 0.972 |
| | Direct | BHV | 0.880 | 0.836 | 0.982 |
| | | Euclid | 0.782 | 0.868 | 0.962 |
| MLE-GTR | KPCA | BHV | 0.506 | 0.522 | 0.590 |
| | | Euclid | 0.504 | 0.778 | 0.824 |
| | t-SNE | BHV | 0.722 | 0.652 | 0.886 |
| | | Euclid | 0.750 | 0.828 | 0.934 |
| | Isomap | BHV | 0.522 | 0.530 | 0.800 |
| | | Euclid | 0.750 | 0.716 | 0.918 |
| | Direct | BHV | 0.612 | 0.620 | 0.736 |
| | | Euclid | 0.536 | 0.762 | 0.680 |
| MLE-HKY | KPCA | BHV | 0.506 | 0.522 | 0.638 |
| | | Euclid | 0.592 | 0.798 | 0.894 |
| | t-SNE | BHV | 0.746 | 0.672 | 0.896 |
| | | Euclid | 0.734 | 0.860 | 0.942 |
| | Isomap | BHV | 0.530 | 0.528 | 0.840 |
| | | Euclid | 0.744 | 0.730 | 0.926 |
| | Direct | BHV | 0.616 | 0.620 | 0.762 |
| | | Euclid | 0.600 | 0.798 | 0.874 |
| MLE-K80 | KPCA | BHV | 0.506 | 0.522 | 0.696 |
| | | Euclid | 0.646 | 0.808 | 0.930 |
| | t-SNE | BHV | 0.728 | 0.614 | 0.914 |
| | | Euclid | 0.730 | 0.876 | 0.942 |
| | Isomap | BHV | 0.530 | 0.530 | 0.866 |
| | | Euclid | 0.740 | 0.742 | 0.946 |
| | Direct | BHV | 0.614 | 0.510 | 0.780 |
| | | Euclid | 0.674 | 0.800 | 0.944 |
| MLE-JC | KPCA | BHV | 0.506 | 0.522 | 0.730 |
| | | Euclid | 0.658 | 0.816 | 0.940 |
| | t-SNE | BHV | 0.738 | 0.690 | 0.908 |
| | | Euclid | 0.754 | 0.874 | 0.946 |
| | Isomap | BHV | 0.526 | 0.528 | 0.876 |
| | | Euclid | 0.734 | 0.736 | 0.950 |
| | Direct | BHV | 0.616 | 0.624 | 0.788 |
| | | Euclid | 0.690 | 0.802 | 0.946 |

Table 1: Comparison: Ncut clustering

| Method | Dim. Red. | Distance | $c$ 0.6 | 0.8 | 1.2 |
|---|---|---|---|---|---|
| NJp | KPCA | BHV | 0.780 | 0.838 | 0.992 |
| | | Euclid | 0.618 | 0.856 | 0.966 |
| | t-SNE | BHV | 0.824 | 0.814 | 0.972 |
| | | Euclid | 0.738 | 0.860 | 0.964 |
| | Isomap | BHV | 0.676 | 0.546 | 0.978 |
| | | Euclid | 0.800 | 0.866 | 0.972 |
| | Direct | BHV | - | - | |
| | | Euclid | 0.782 | 0.840 | 0.966 |
| MLE-GTR | KPCA | BHV | 0.582 | 0.618 | 0.742 |
| | | Euclid | 0.504 | 0.782 | 0.856 |
| | t-SNE | BHV | 0.744 | 0.612 | 0.870 |
| | | Euclid | 0.718 | 0.832 | 0.934 |
| | Isomap | BHV | 0.602 | 0.564 | 0.804 |
| | | Euclid | 0.748 | 0.766 | 0.922 |
| | Direct | BHV | - | - | |
| | | Euclid | 0.548 | 0.734 | 0.682 |
| MLE-HKY | KPCA | BHV | 0.608 | 0.622 | 0.750 |
| | | Euclid | 0.574 | 0.802 | 0.924 |
| | t-SNE | BHV | 0.688 | 0.664 | 0.908 |
| | | Euclid | 0.726 | 0.848 | 0.946 |
| | Isomap | BHV | 0.592 | 0.564 | 0.846 |
| | | Euclid | 0.740 | 0.766 | 0.928 |
| | Direct | BHV | - | - | |
| | | Euclid | 0.572 | 0.774 | 0.910 |
| MLE-K80 | KPCA | BHV | 0.584 | 0.626 | 0.778 |
| | | Euclid | 0.650 | 0.808 | 0.936 |
| | t-SNE | BHV | 0.746 | 0.622 | 0.916 |
| | | Euclid | 0.668 | 0.884 | 0.940 |
| | Isomap | BHV | 0.590 | 0.562 | 0.856 |
| | | Euclid | 0.752 | 0.774 | 0.946 |
| | Direct | BHV | - | - | |
| | | Euclid | 0.696 | 0.796 | 0.950 |
| MLE-JC | KPCA | BHV | 0.614 | 0.618 | 0.788 |
| | | Euclid | 0.676 | 0.810 | 0.940 |
| | t-SNE | BHV | 0.700 | 0.644 | 0.912 |
| | | Euclid | 0.750 | 0.880 | 0.946 |
| | Isomap | BHV | 0.594 | 0.562 | 0.876 |
| | | Euclid | 0.746 | 0.714 | 0.950 |
| | Direct | BHV | - | - | |
| | | Euclid | 0.702 | 0.798 | 0.946 |

Table 2: Comparison: $k$-means clustering

| Method | Dim. Red. | Distance | $c$ 0.6 | 0.8 | 1.2 |
|---|---|---|---|---|---|
| NJp | KPCA | BHV | 0.508 | 0.502 | 0.542 |
| | | Euclid | 0.540 | 0.664 | 0.572 |
| | t-SNE | BHV | 0.546 | 0.692 | 0.976 |
| | | Euclid | 0.610 | 0.834 | 0.950 |
| | Isomap | BHV | 0.534 | 0.574 | 0.890 |
| | | Euclid | 0.506 | 0.502 | 0.974 |
| | Direct | BHV | 0.510 | 0.502 | 0.522 |
| | | Euclid | 0.524 | 0.502 | 0.560 |
| MLE-GTR | KPCA | BHV | 0.504 | 0.572 | 0.572 |
| | | Euclid | 0.522 | 0.648 | 0.568 |
| | t-SNE | BHV | 0.636 | 0.668 | 0.824 |
| | | Euclid | 0.648 | 0.700 | 0.932 |
| | Isomap | BHV | 0.514 | 0.544 | 0.516 |
| | | Euclid | 0.502 | 0.582 | 0.524 |
| | Direct | BHV | 0.504 | 0.548 | 0.508 |
| | | Euclid | 0.508 | 0.502 | 0.502 |
| MLE-HKY | KPCA | BHV | 0.504 | 0.548 | 0.554 |
| | | Euclid | 0.604 | 0.612 | 0.576 |
| | t-SNE | BHV | 0.652 | 0.620 | 0.910 |
| | | Euclid | 0.568 | 0.802 | 0.942 |
| | Isomap | BHV | 0.538 | 0.546 | 0.510 |
| | | Euclid | 0.532 | 0.512 | 0.578 |
| | Direct | BHV | 0.502 | 0.508 | 0.508 |
| | | Euclid | 0.508 | 0.510 | 0.502 |
| MLE-K80 | KPCA | BHV | 0.504 | 0.520 | 0.554 |
| | | Euclid | 0.598 | 0.572 | 0.550 |
| | t-SNE | BHV | 0.652 | 0.648 | 0.832 |
| | | Euclid | 0.670 | 0.858 | 0.942 |
| | Isomap | BHV | 0.514 | 0.544 | 0.734 |
| | | Euclid | 0.550 | 0.552 | 0.540 |
| | Direct | BHV | 0.510 | 0.508 | 0.514 |
| | | Euclid | 0.510 | 0.502 | 0.502 |
| MLE-JC | KPCA | BHV | 0.504 | 0.544 | 0.554 |
| | | Euclid | 0.540 | 0.654 | 0.556 |
| | t-SNE | BHV | 0.650 | 0.642 | 0.838 |
| | | Euclid | 0.658 | 0.678 | 0.944 |
| | Isomap | BHV | 0.538 | 0.544 | 0.740 |
| | | Euclid | 0.506 | 0.628 | 0.582 |
| | Direct | BHV | 0.512 | 0.508 | 0.514 |
| | | Euclid | 0.512 | 0.508 | 0.536 |

Table 3: Comparison: hierarchical clustering (average linkage)

# 3 Experiments on dimension reduction

Several methods for dimension reduction were applied to the simulated data discussed in Section 3.1. Isomap (Tenenbaum *et al.*, 2000), Laplacian eigenmap (Belkin and Niyogi, 2001), KPCA, and t-SNE were applied to the distance matrix of the BHV tree space and the Euclidean distance matrix assuming the cone space. As tree reconstruction methods, NJp, MLE-GTR, MLE-HKY, and MLE-K80 were used. The results for $c = 0.8$ and 1.2 (ratio in the species depth) are shown. The blue and red colors respectively indicate the genes in each of the two species trees.

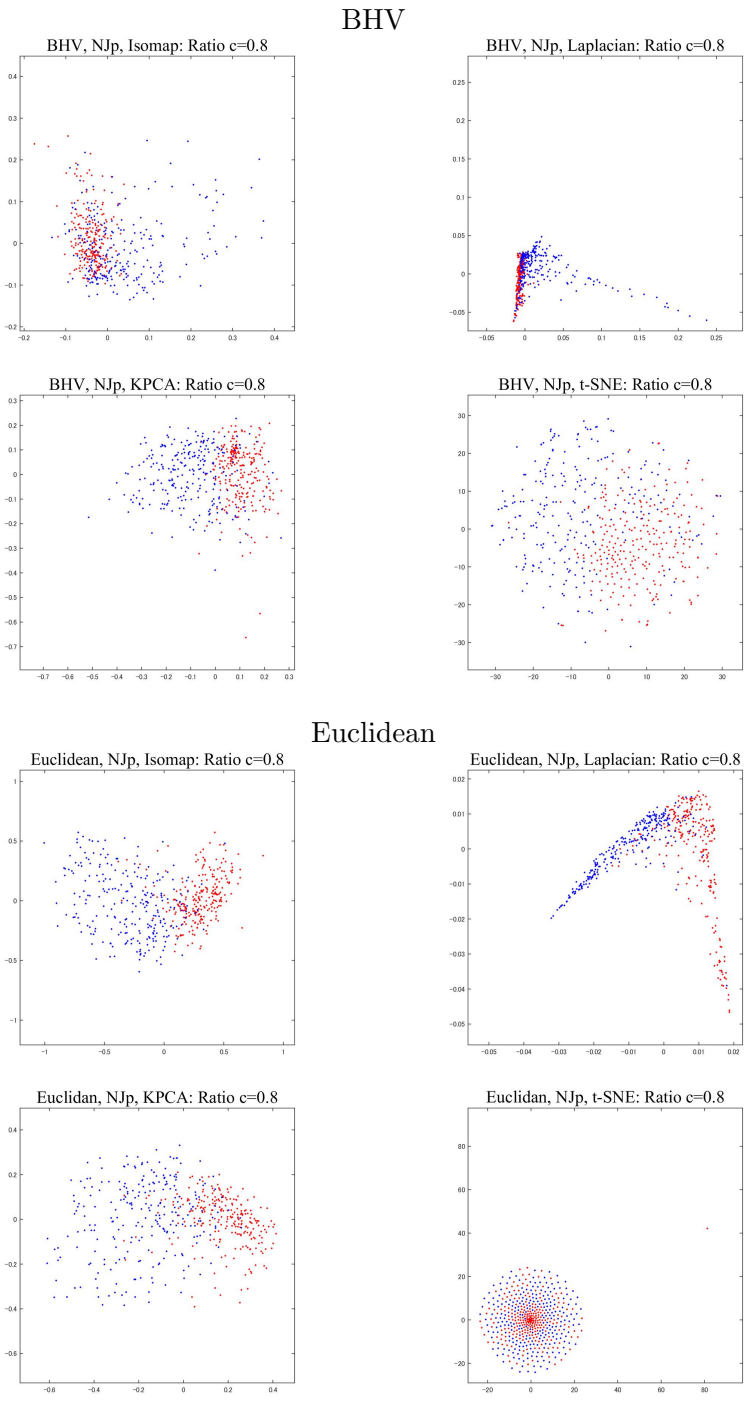


Figure 4: Dimension reduction, NJp, $c = 0.8$ (upper half: BHV, lower half: Euclidean)

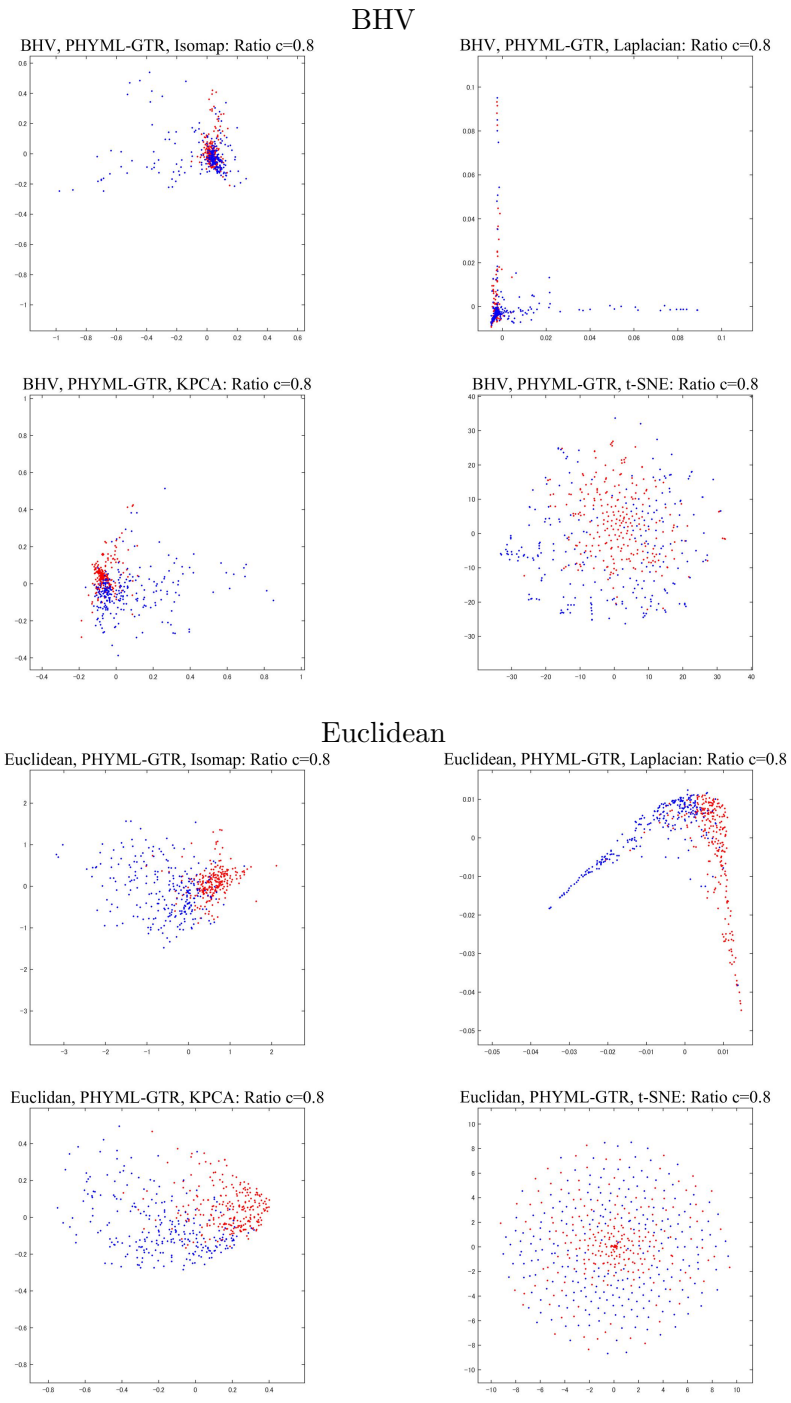


Figure 5: Dimension reduction, ML-GTR, $c = 0.8$ (upper half: BHV, lower half: Euclidean)

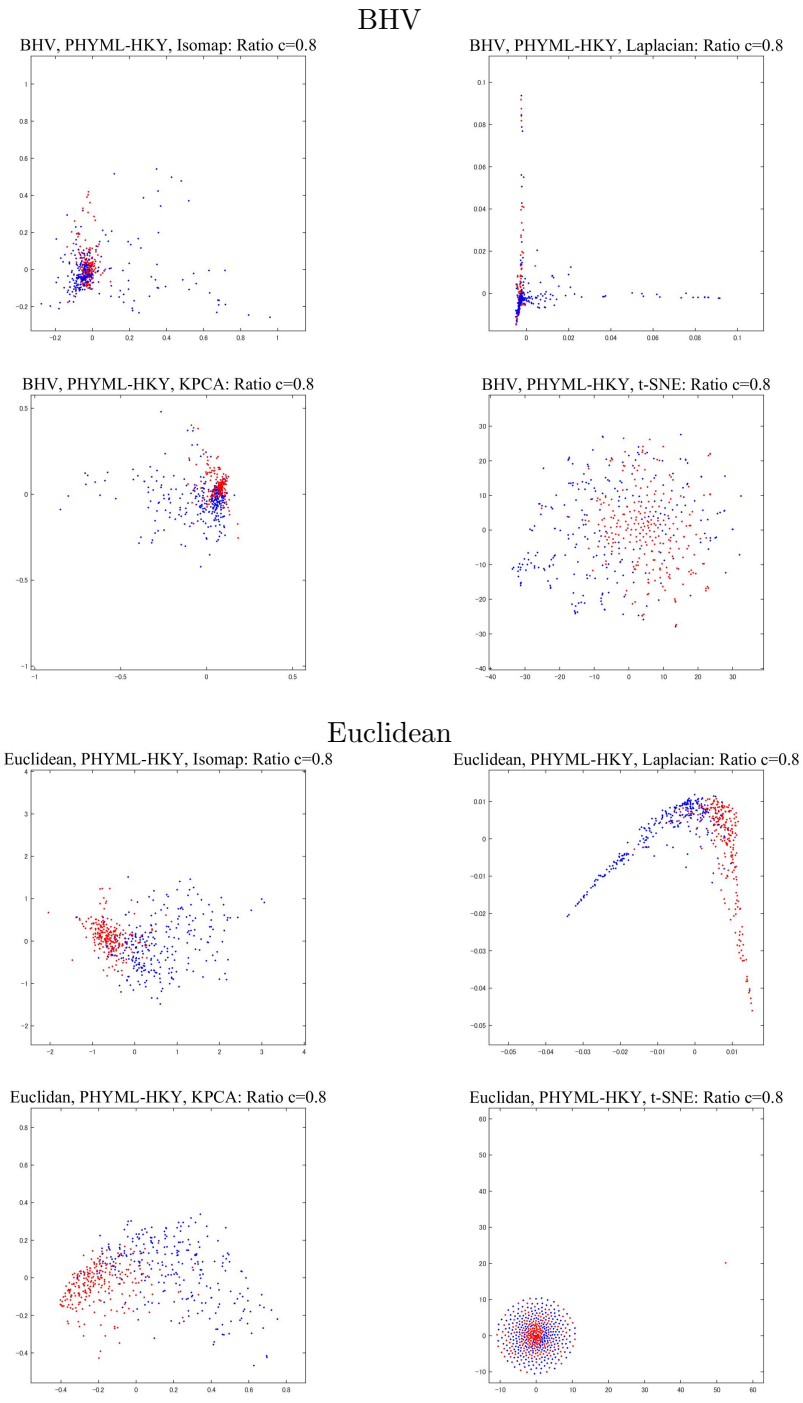


Figure 6: Dimension reduction, ML-HKY, $c = 0.8$ (upper half: BHV, lower half: Euclidean)

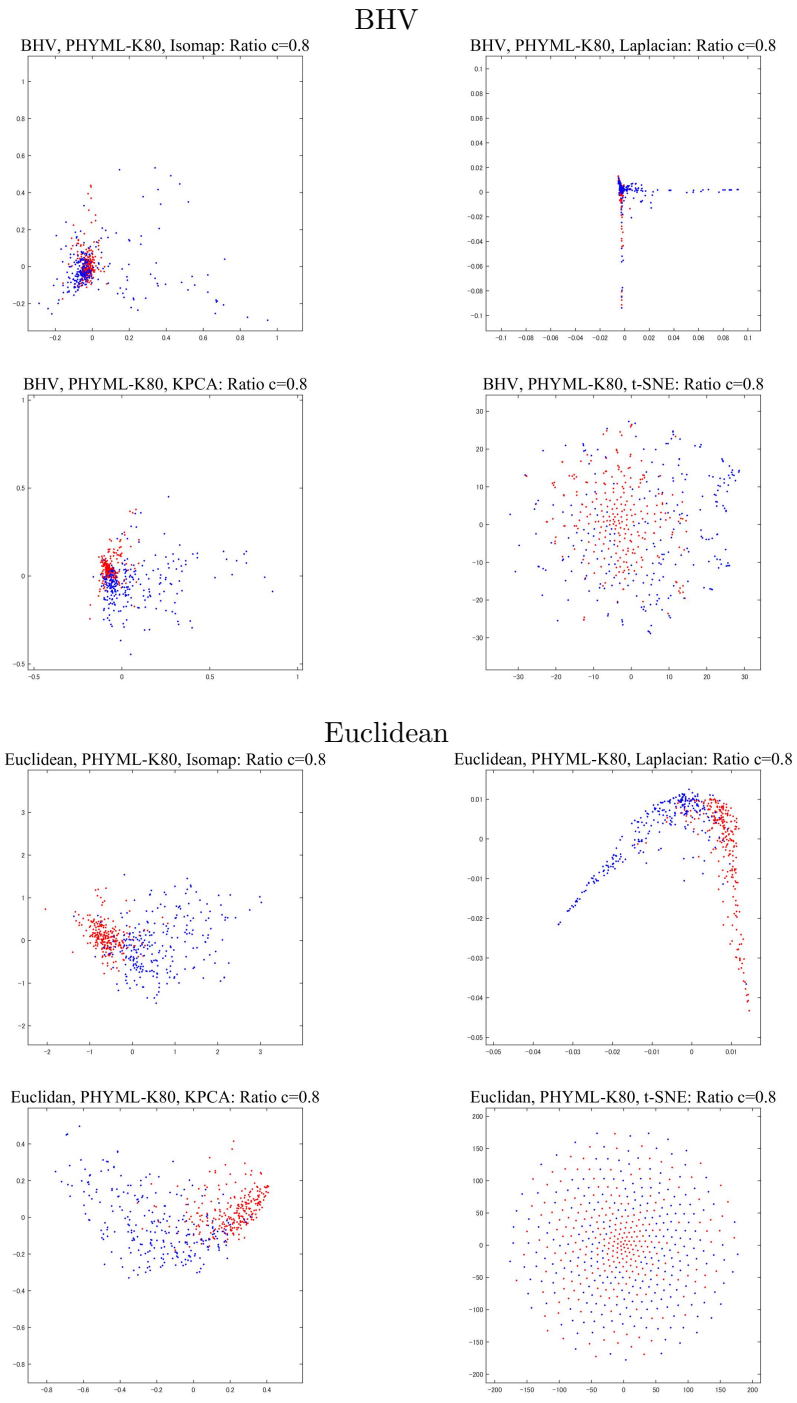


Figure 7: Dimension reduction, ML-K80, $c = 0.8$ (upper half: BHV, lower half: Euclidean)

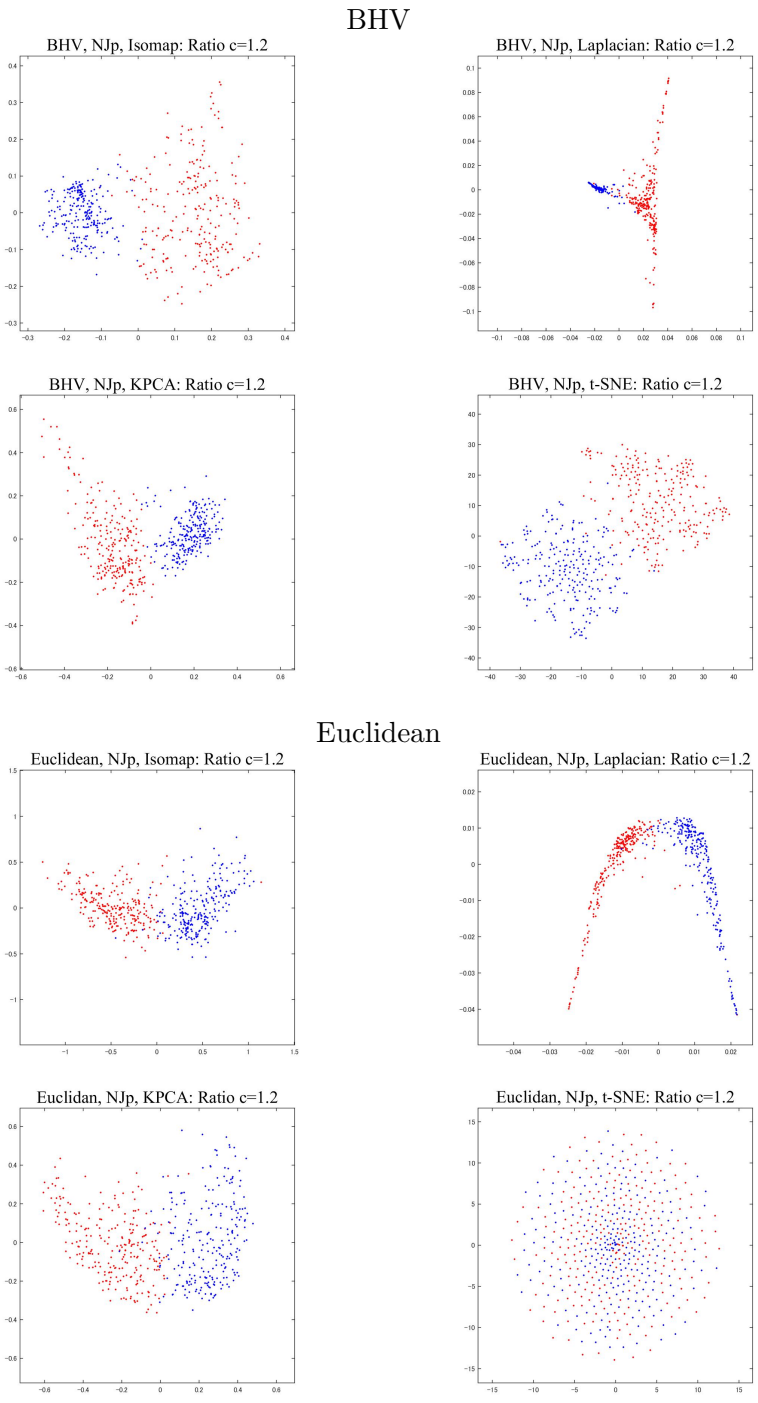


Figure 8: Dimension reduction, NJp, $c = 1.2$ (upper half: BHV, lower half: Euclidean)

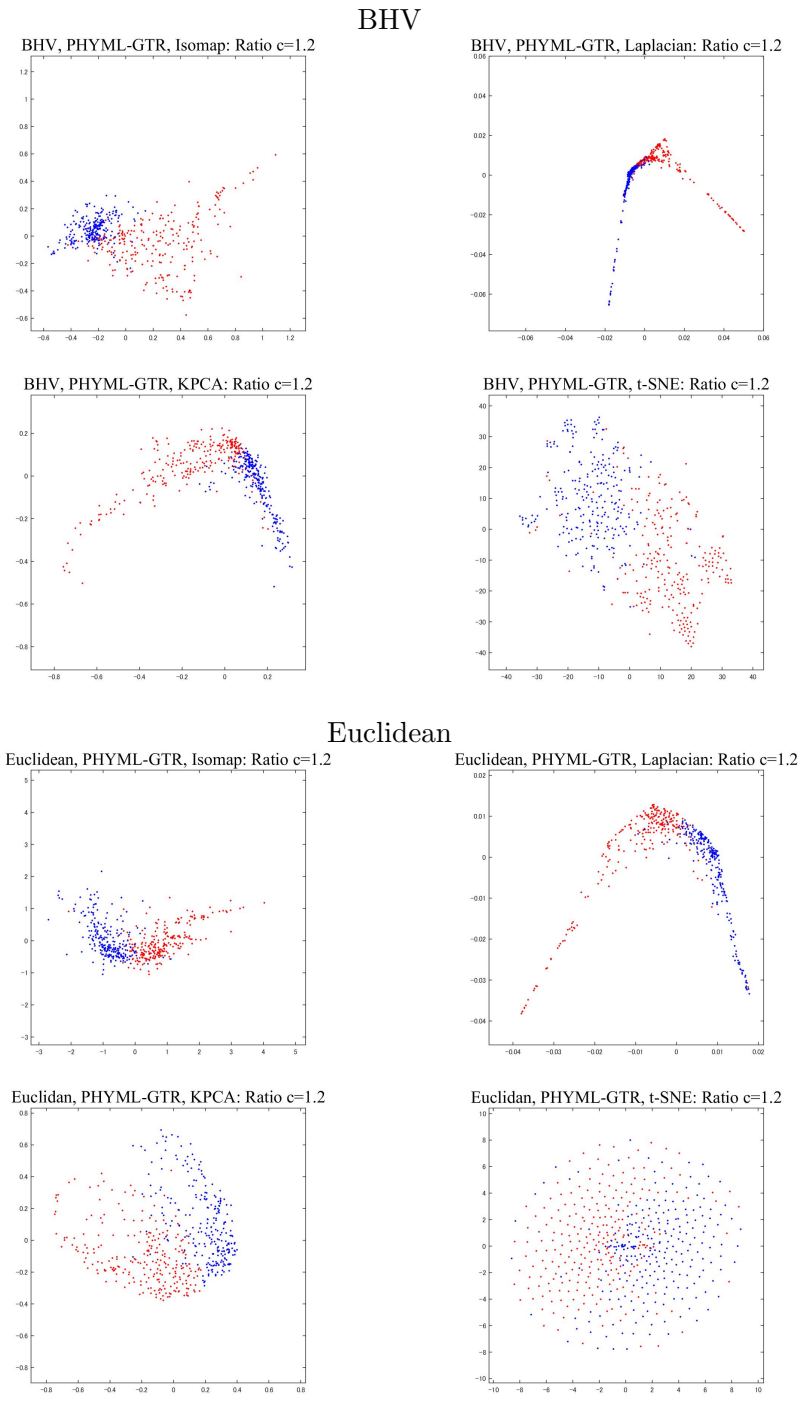


Figure 9: Dimension reduction, ML-GTR, $c = 1.2$ (upper half: BHV, lower half: Euclidean)

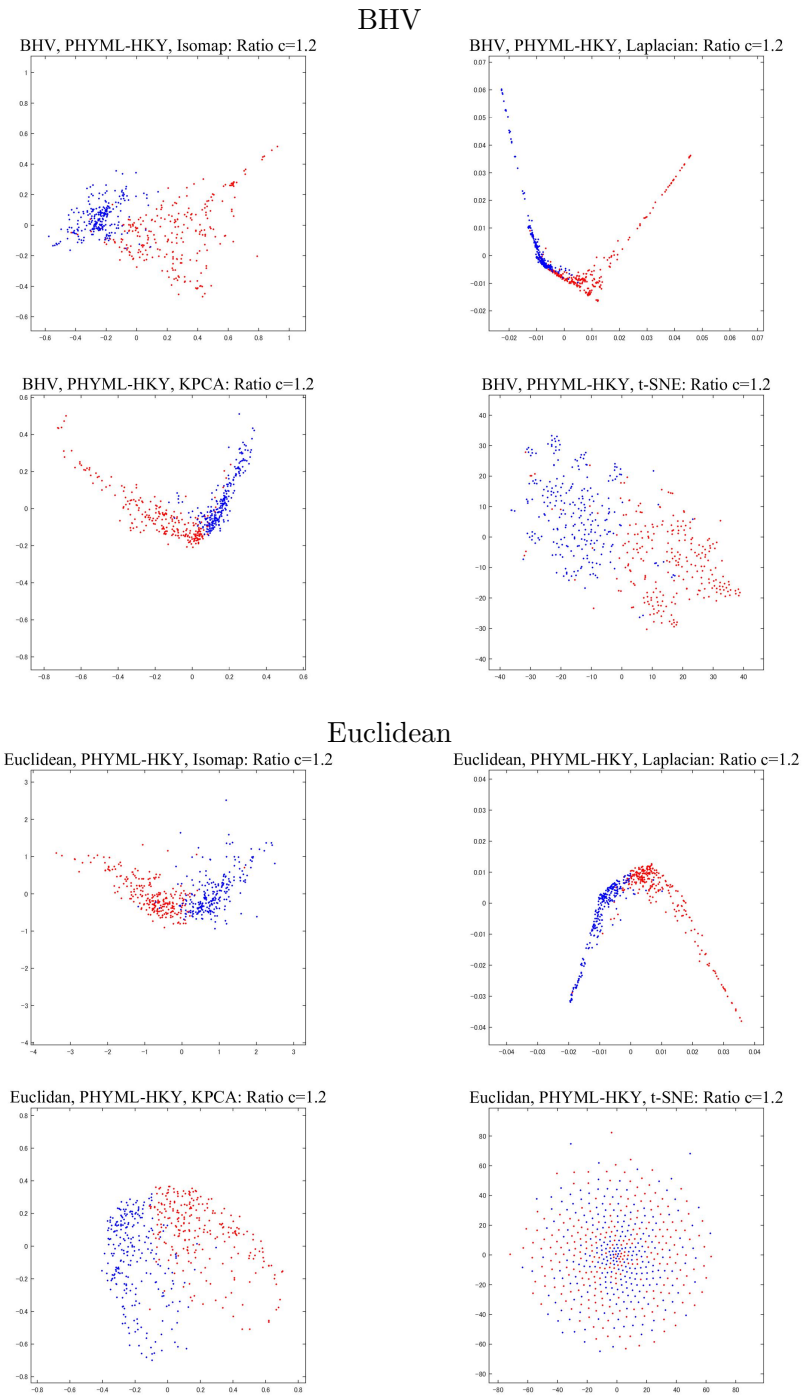


Figure 10: Dimension reduction, ML-HKY, $c = 1.2$ (upper half: BHV, lower half: Euclidean)

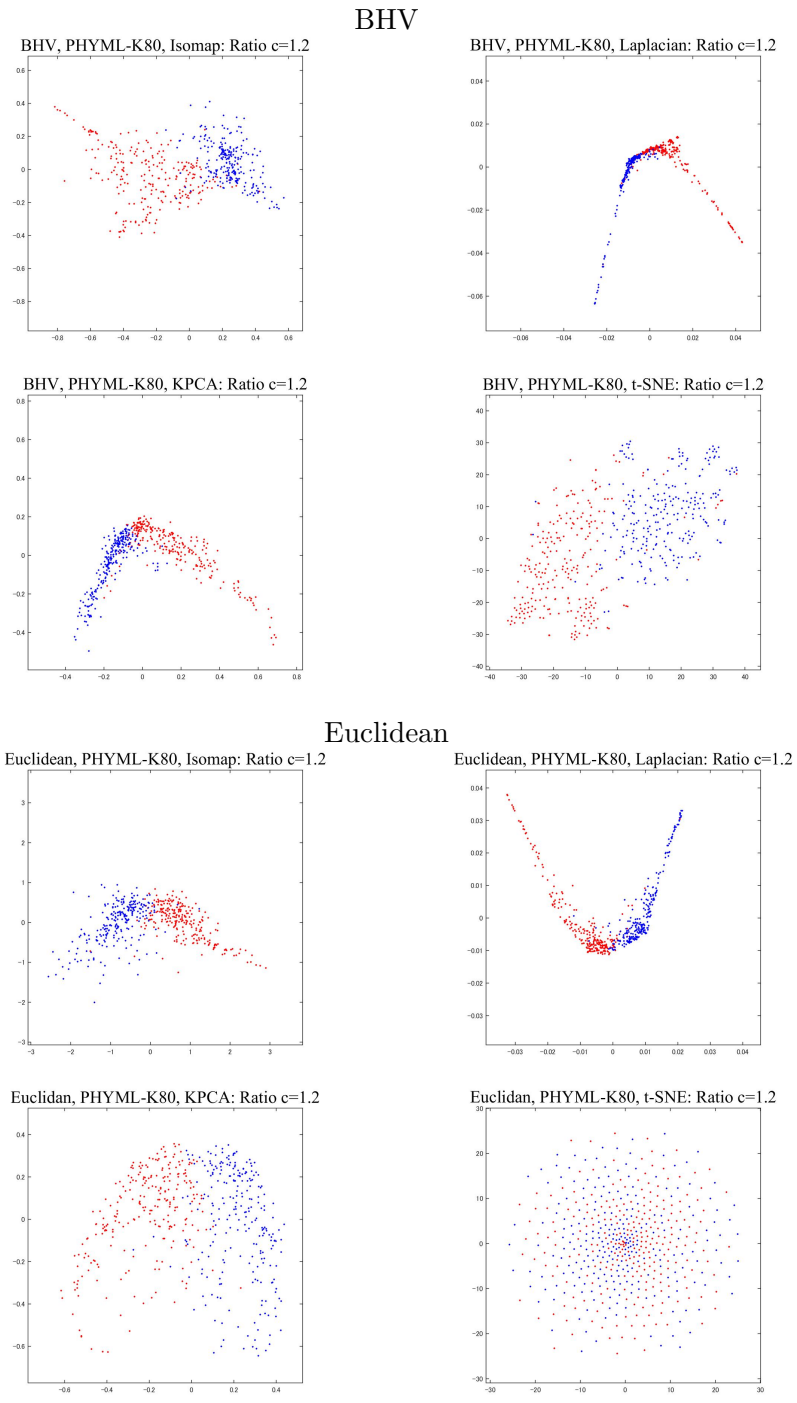


Figure 11: Dimension reduction, ML-K80, $c = 1.2$ (upper half: BHV, lower half: Euclidean)

# References


E.D. Amir, K.L. Davis, M.D. Tadmor, E.F. Simonds, J.H. Levine, S.C. Bendall, D.K. Shenfeld, S. Krishnaswamy, G.P. Nolan, and D. Pe'er. viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nature Biotechnology*, 31(6):545–552, 2013.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems*, 14:586–691, 2001.

A. Platzer. Visualization of SNPs with t-SNE. *PLOS ONE*, 2013.

A.A. Popescu, A.L. Harper, Trick M., Bancroft I., and Huber K.T. A novel and fast approach for population structure inference using kernel-pca and optimization. *Genetics*, 198(4):1421–1431, 2014.

F. Reverter, E. Vegas, and J.M. Oller. Kernel-PCA data integration with enhanced interpretability. *BMC Systems Biology*, 8((Suppl 2)):S6, 2014.

A. Saadatpour, G. Guo, S.H. Orkin, and G.-C. Yuan. Characterizing heterogeneity in leukemic cells using single-cell gene expression analysis. *Genome Biology*, 15(12):525, 2014.

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

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

J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

L.J.P. van der Maaten and G.E. Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

L.J.P. van der Maaten. Accelerating t-SNE using tree-based algorithms. *Journal of Machine Learning Research*, 15:3221–3245, 2014.