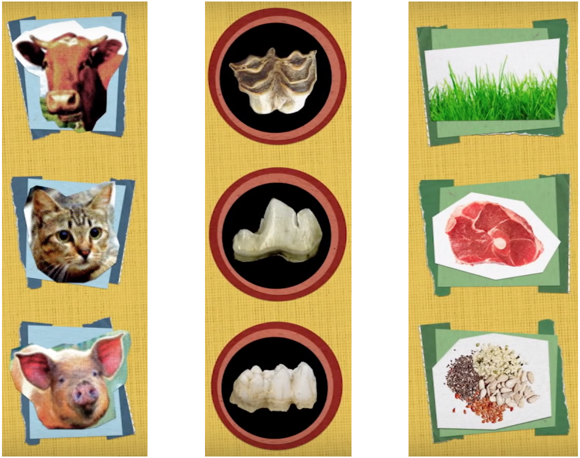
Math and statistics in teeth and bones

What animals are these?

What animals are these?

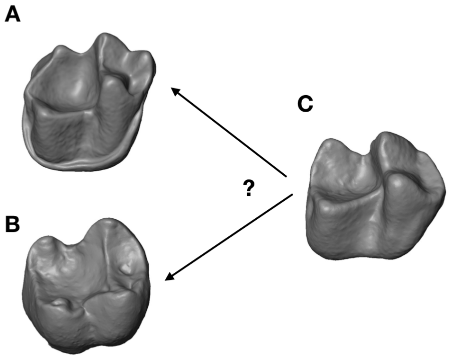
What animals are these?

What animals are these?



Bio question 1: how to define features on teeth to infer diet?

Which one is similar?

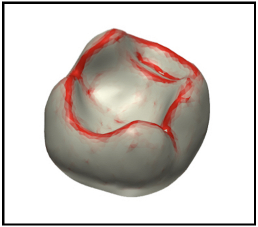


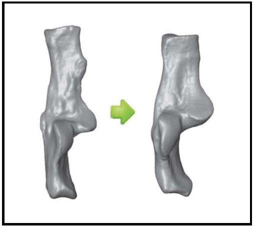
Bio question 2: how to define distance between a pair of shapes?

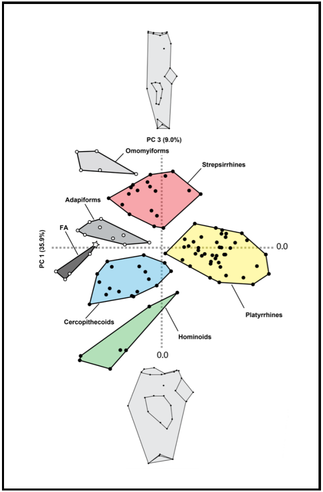
Is there a pattern in the collection?

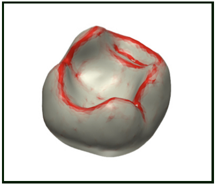
Bio question 3: how to visualize the shape space?

Math and statistics in teeth and bones







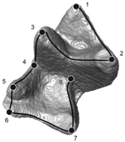


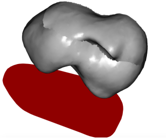
Part I:

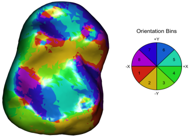
features on individual shape

Features on a tooth

Traditionally,

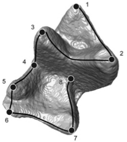


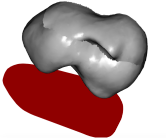


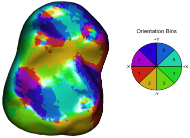


Features on a tooth

Traditionally,







More recently,

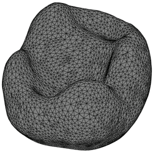
Dirichlet energy $= \int_{\mathcal{S}}$  $d\mathcal{S}$

(change in the normal direction)

Bunn, J. M., Boyer, D. M., Lipman, Y., St. Clair, E. M., Jernvall, J., & Daubechies, I. (2011). Comparing Dirichlet normal surface energy of tooth crowns, a new technique of molar shape quantification for dietary inference, with previous methods in isolation and in combination. American Journal of Physical Anthropology, 145(2), 247-261.

Compute on a discrete surface

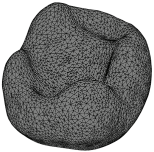
Triangles stitched to build a surface



A mesh consists of triangular faces and points.

Compute on a discrete surface

Triangles stitched to build a surface



A mesh consists of triangular faces and points.

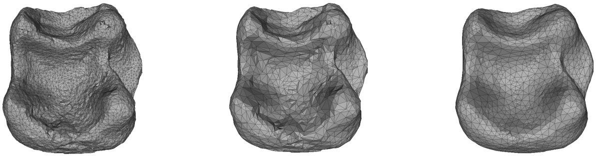
$$\text{Dirichlet energy} \approx \sum_{\Delta \in \text{ Faces}}$$

$$\text{Area}(\Delta)$$

(change in normal direction of the face Δ)

The implementation is sensitive

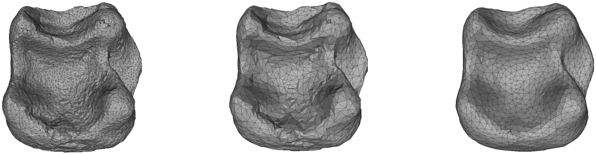
In practice, we have



all representing the same shape

The implementation is sensitive

In practice, we have



all representing the same shape

 gives very different numbers.

A robust algorithm

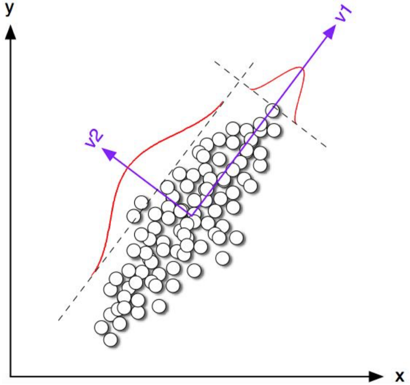
Key statistics observation:

$$\text{Dirichlet energy } = \int_{\mathcal{S}} \text{“curvature”}\ d\mathcal{S}$$

We compute curvature in a more stable way via the Principal Component Analysis (PCA).

Principal component analysis (PCA)

... finds the "shape" or big pattern of data.



Principal component

Principal component score

PCA algorithm

1. Given $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ a $d \times n$ matrix construct

$$\widehat{\Sigma} = \frac{1}{n-1}(X - \bar{X})(X - \bar{X})^\intercal$$

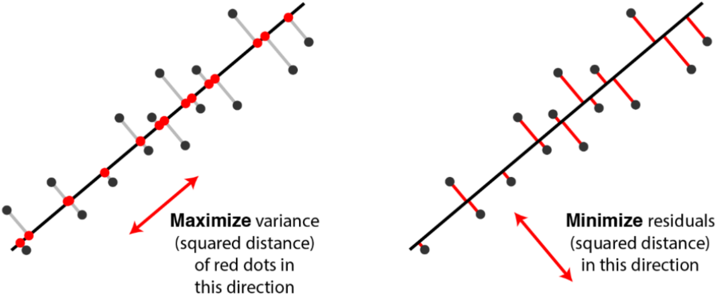
2. Eigen-decomposition of $\hat{\Sigma}$

$$\widehat{\Sigma} v_i = \lambda_i v_i, \lambda_1 \geq \lambda_2 \ldots, \geq \lambda_d$$

v_i: i-th principal component
λ_i: i-th principal component score

PCA fits a line in 2D

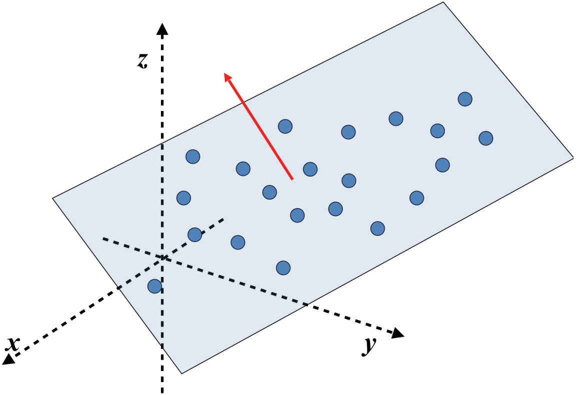
v_1 gives you a line that



Two equivalent views of principal component analysis.

PCA fits a plane in 3D

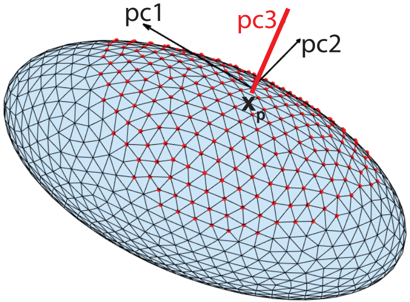
span$\{v_1, v_2\}$ gives you a plane



λ_3, variability in the v_3 direction approximates
how much a surface bends, therefore curvature.

Estimate curvature at a point x_p with PCA

Apply PCA to a small neighborhood of points around x_p



Estimate curvature at a point x_p with PCA

Apply PCA to a small neighborhood of points around x_p

tangent plane

$$\approx \text{span}\{v_1, v_2\}$$

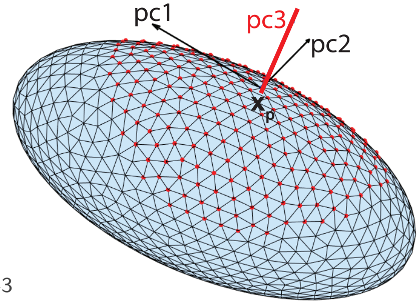
normal vector

$$\approx v_3$$

approximate curvature

denoted 

$$\approx \frac{\lambda_3}{\lambda_1+\lambda_2+\lambda_3},\ \lambda_1 \geq \lambda_2 \geq \lambda_3$$



Estimate curvature at a point x_p with PCA

Apply PCA to a small neighborhood of points around x_p

tangent plane

$$\approx \text{span}\{v_1, v_2\}$$

normal vector

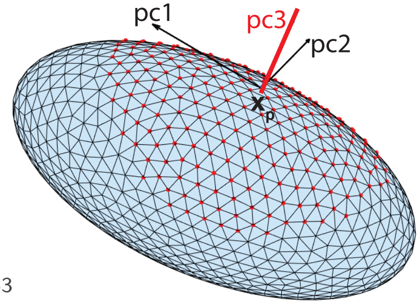
$$\approx v_3$$

approximate curvature

denoted 

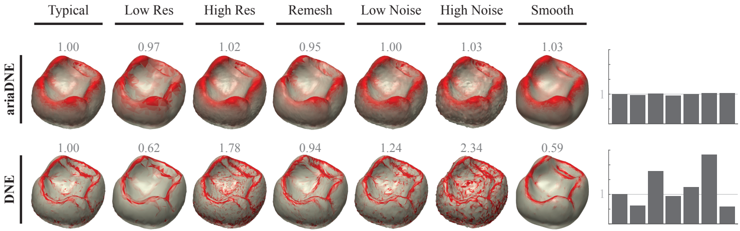
$$\approx \frac{\lambda_3}{\lambda_1+\lambda_2+\lambda_3}, \ \lambda_1 \geq \lambda_2 \geq \lambda_3$$

Dirichlet energy $\approx \sum_p$  $w(p)$



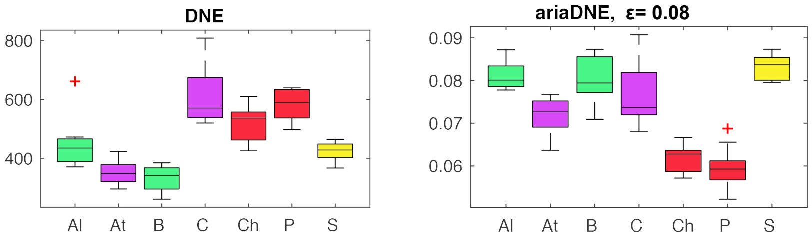
Effects of changing discrete representation

We perturb the shape by ...



red indicates high  (top) and  (bottom)

Power of differentiating species by diet

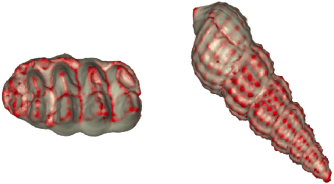


Color indicates dietary preferences:

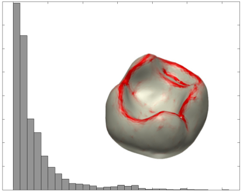
green – leaf; purple – fruit;
red – nut; yellow – insect.

Some questions to ponder

- "Complexity" on other shapes?



- Better shape descriptor with the curvature field e.g., mean? variance?



Part II:

distance between shapes

Distance $d : (\cdot, \cdot) \to \mathbb{R}$

Distance $d : (\cdot, \cdot) \to \mathbb{R}$

$d(4135382162, 4135382161)$

Distance $d : (\cdot, \cdot) \to \mathbb{R}$

$d(4135382162, 4135382161)$

$$1$$

Distance $d : (\cdot, \cdot) \to \mathbb{R}$

$d(4135382162, 4135382161)$

$$1$$

$d\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right)$

$$\sqrt{2}$$

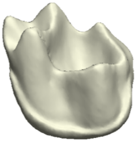
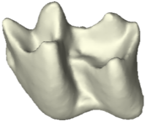
Distance $d : (\cdot, \cdot) \to \mathbb{R}$

$$d(4135382162, 4135382161)$$

$$1$$

$$d\left(\begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}\right)$$

$$\sqrt{2}$$

d ( , )

?

Equivalent shapes

1. rotation

$d\Big($  ,  $\Big) = 0$

2. translation

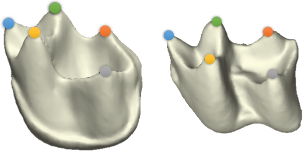
$d\Big($  ,  $\Big) = 0$

3. scale

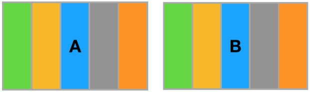
$d\Big($  ,  $\Big) = 0$

Traditional procrustes distance

1. Center shapes at origin and scale to unit size
2. Put n landmarks on shapes

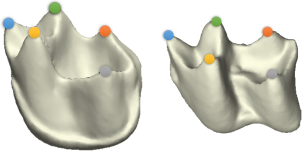


3. Write as 3 by n matrix

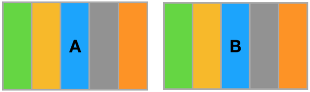


Traditional procrustes distance

1. Center shapes at origin and scale to unit size
2. Put n landmarks on shapes



3. Write as 3 by n matrix

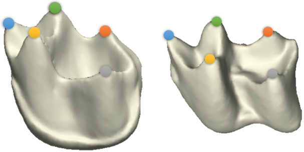


4. Find rotation $\mathbf{R}$ (3 by 3) minimizing

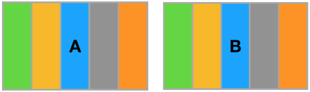
$$d(\mathbf{R}A, B)$$

Traditional procrustes distance

1. Center shapes at origin and scale to unit size
2. Put n landmarks on shapes



3. Write as 3 by n matrix



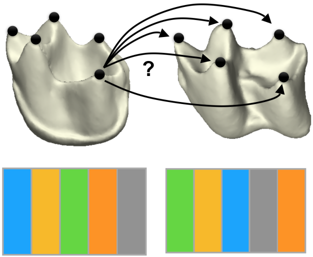
4. Find rotation $\mathbf{R}$ (3 by 3) minimizing

$$d(\mathbf{R}A, B)$$

Putting landmarks is time-consuming and subjective!

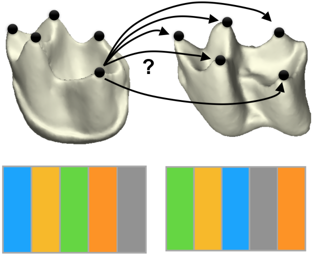
Modified procrustes distance

Key math observation:
Landmarking is ordering points. Points are given as columns in a matrix. Reordering the columns is right-multiplying a permutation matrix.



Modified procrustes distance

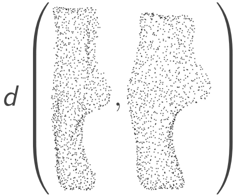
Key math observation:
Landmarking is ordering points. Points are given as columns in a matrix. Reordering the columns is right-multiplying a permutation matrix.



Find permutation **P** (n by n) and **R** minimizing

$$d(\mathbf{R}A, B\mathbf{P})$$

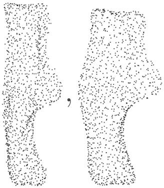
An fully automated algorithm



1. subsample: spread n points evenly on each surface
2. iterative search:
   find $\mathbf{R}$ minimizing $d(\mathbf{R}A, BP)$
   until converges

Boyer, D.M., Puente, J., Gladman, J.T., Glynn, C., Mukherjee, S., Yapuncich, G.S. and Daubechies, I., (2015). A new fully automated approach for aligning and comparing shapes. The Anatomical Record, 298(1), 249-276.

An fully automated algorithm

$d\Bigg($  $\Bigg)$

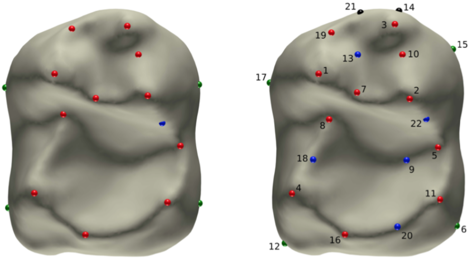
1. subsample: spread n points evenly on each surface
2. iterative search:
   find **P** minimizing $d(RA, B\mathbf{P})$
   until converges

Boyer, D.M., Puente, J., Gladman, J.T., Glynn, C., Mukherjee, S., Yapuncich, G.S. and Daubechies, I., (2015).
A new fully automated approach for aligning and comparing shapes.
The Anatomical Record, 298(1), 249-276.

How to select the points and how many?

larger $n \rightarrow$ more accuracy
but computationally more expensive

If a computer can place landmarks like a biologist,
small n could still give us accurate results.

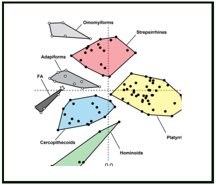


Gao, T., Kovalsky, S.Z. and Daubechies, I., 2019. Gaussian process landmarking on manifolds. SIAM Journal on Mathematics of Data Science, 1(1), pp.208-236.

Stephen, A., 2019. Optimizing bioinformatic analyses of 3D anatomical data. Senior Thesis, Duke University.

Undergraduate projects

- Test existing shape matching algorithms
- Design new distance for shapes
- Align multiple shapes
- Develop open-source software for scientific research

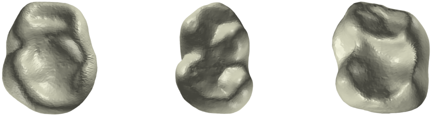


Part III:

patterns in the shape space

The space of shapes

3 teeth
(data objects)



discretized and
represented in $\mathbb{R}^{5000}$
(data points)

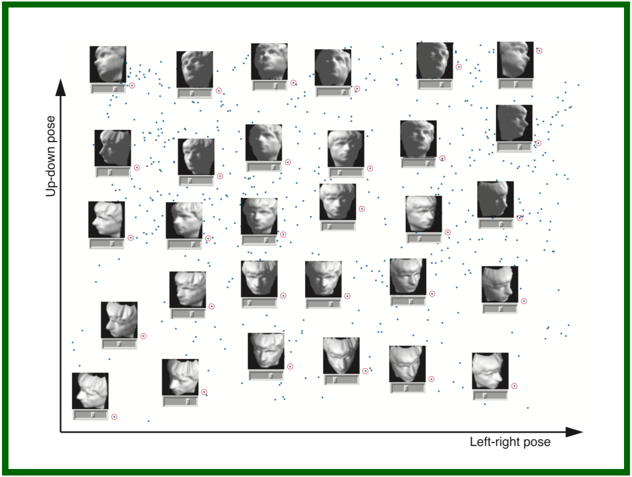
$$\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_{5000} \end{bmatrix} \quad \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ \vdots \\ w_{5000} \end{bmatrix} \quad \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ \vdots \\ v_{5000} \end{bmatrix}$$

Find a low dimensional representation – dimension reduction!

Dimension reduction

embedded data

$\mathbb{R}^{\text{large}}$



$\mathbb{R}^2$

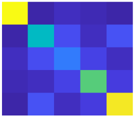
Diffusion maps are easy to compute

1. Compute pairwise distance

$$d_{ij}$$

2. Build diffusion matrix $K^{(t)}$

$$[K_{ij}^{(t)}] = \exp\left(-d_{ij}^2/t\right)$$



Normalize each row to sum to 1

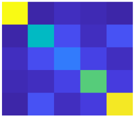
Diffusion maps are easy to compute

1. Compute pairwise distance

$$d_{ij}$$

2. Build diffusion matrix $K^{(t)}$

$$[K_{ij}^{(t)}] = \exp\left(-d_{ij}^2/t\right)$$



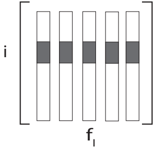
Normalize each row to sum to 1

3. Eigen-decompose $K^{(t)}$

$$K^{(t)} f_l = \lambda_l f_l$$

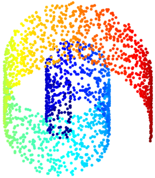
4. Diffusion maps is

$$x_i \to (f_1^{(i)}, f_2^{(i)})$$



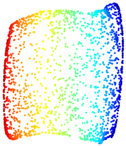
Diffusion maps are hard to tune

Original data: swiss roll

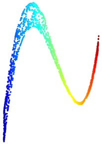


Embedded data with different t

(a) t too large



(b) "right" t



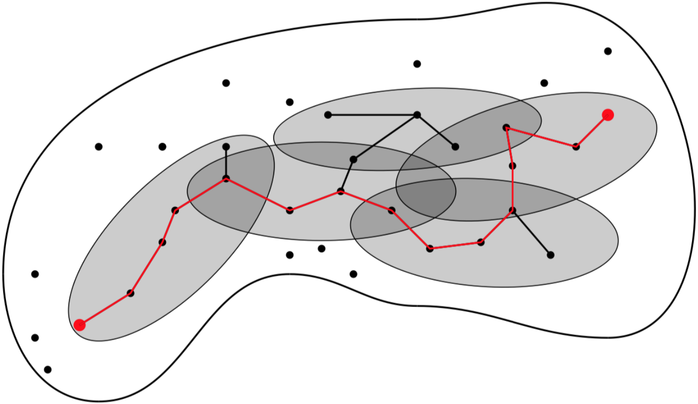
(c) t too small

Diffusion process

Macro – ink spread out over time.
Micro – sum of large number of small random forces
by molocules doing random walk

Random walk on data

...knit together local geometry



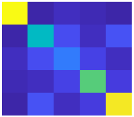
Diffusion maps are easy to compute

1. Compute pairwise distance

$$d_{ij}$$

2. Build diffusion matrix $K^{(t)}$

$$[K_{ij}^{(t)}] = \exp\left(-d_{ij}^2/t\right)$$



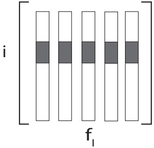
Normalize each row to sum to 1

3. Eigen-decompose $K^{(t)}$

$$K^{(t)} f_l = \lambda_l f_l$$

4. Diffusion maps is

$$x_i \to (f_1^{(i)}, f_2^{(i)})$$



Can you hear the shape of a drum?

Eigenvalues and eigenfunctions of diffusion operator can tell you a lot about the geometry of the domain.

Diffusion maps come from diffusion operators

Key math observation:

$$K^{(t)} \xrightarrow{\text{more data, small } t} e^{-t\Delta}$$

Diffusion maps come from diffusion operators

Key math observation:

$$K^{(t)} \xrightarrow{\text{more data, small } t} e^{-t\Delta}$$

semigroup properties of diffusion op.

$$e^{-(t_1+t_2)\Delta} = e^{-t_1\Delta} e^{-t_2\Delta}$$

with discrete approximation

$$K^{(t_1+t_2)} \approx K^{(t_1)} K^{(t_2)}$$

Diffusion maps come from diffusion operators

Key math observation:

$$K^{(t)} \xrightarrow{\text{more data, small } t} e^{-t\Delta}$$

semigroup properties of diffusion op.

$$e^{-(t_1+t_2)\Delta} = e^{-t_1\Delta}e^{-t_2\Delta}$$

with discrete approximation

$$K^{(t_1+t_2)} \approx K^{(t_1)}K^{(t_2)}$$

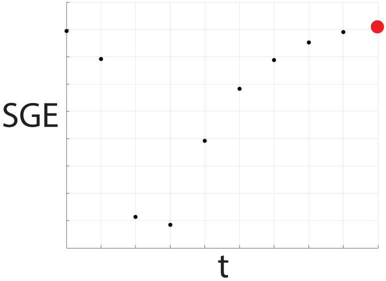
...this gives a semi-group test for choosing t.

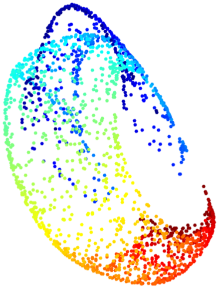
Semi-group test is simple and effective

Semi-group error (SGE)

$$SGE = \left\| \underbrace{K^{(2t)}}_{\left[e^{-\frac{\|\cdot\|^2}{2t}} \right]} - \underbrace{K^{(t)} \cdot K^{(t)}}_{\left[e^{-\frac{\|\cdot\|^2}{t}} \right]^2} \right\|$$

The right t should "minimize" SGE.



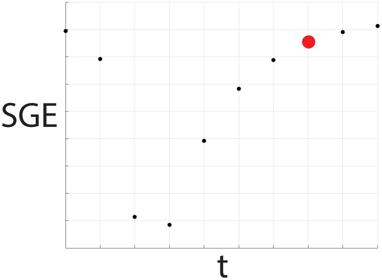


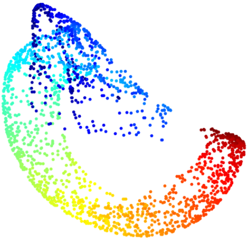
Semi-group test is simple and effective

Semi-group error (SGE)

$$SGE = \Big\| \underset{\left[\, e^{-\frac{\|\cdot\|^2}{2t}} \, \right]}{K^{(2t)}} \; - \; \underset{\left[\, e^{-\frac{\|\cdot\|^2}{t}} \, \right]}{K^{(t)} \cdot K^{(t)}} \Big\|^2$$

The right t should “minimize” SGE.



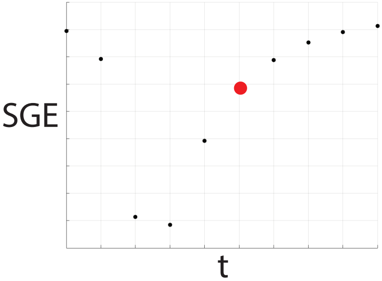


Semi-group test is simple and effective

Semi-group error (SGE)

$$SGE = \Big\| \underbrace{K^{(2t)}}_{\left[\, e^{-\frac{\|\cdot\|^2}{2t}} \,\right]} - \underbrace{K^{(t)} \cdot K^{(t)}}_{\left[\, e^{-\frac{\|\cdot\|^2}{t}} \,\right]^2} \Big\|$$

The right t should "minimize" SGE.



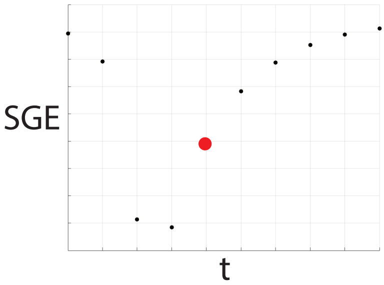
Semi-group test is simple and effective

Semi-group error (SGE)

$$SGE = \left\| \quad K^{(2t)} \quad - \quad K^{(t)} \cdot K^{(t)} \quad \right\|$$

$$\left[\quad e^{-\frac{\|\cdot\|^2}{2t}} \quad \right] \quad \left[\quad e^{-\frac{\|\cdot\|^2}{t}} \quad \right]^2$$

The right t should "minimize" SGE.

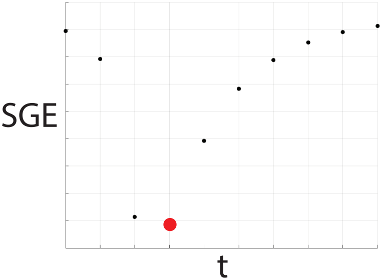


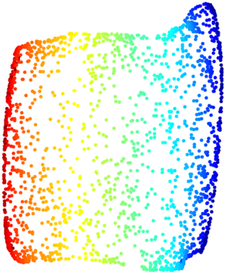
Semi-group test is simple and effective

Semi-group error (SGE)

$$SGE = \| \quad \underset{\left[\; e^{-\frac{\|\cdot\|^2}{2t}} \;\right]}{K^{(2t)}} \quad - \quad \underset{\left[\; e^{-\frac{\|\cdot\|^2}{t}} \;\right]^2}{K^{(t)} \cdot K^{(t)}} \quad \|$$

The right t should "minimize" SGE.



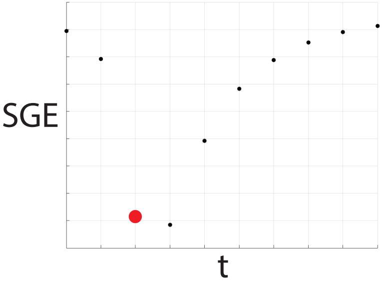


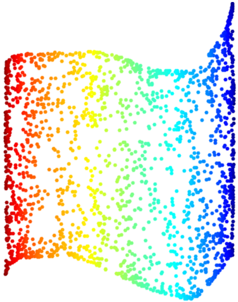
Semi-group test is simple and effective

Semi-group error (SGE)

$$SGE = \left\| \underset{\left[\; e^{-\frac{||\cdot||^2}{2t}} \;\right]}{K^{(2t)}} - \underset{\left[\; e^{-\frac{||\cdot||^2}{t}} \;\right]}{K^{(t)} \cdot K^{(t)}} \right\|^2$$

The right t should "minimize" SGE.





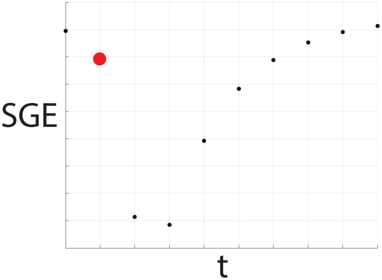
Semi-group test is simple and effective

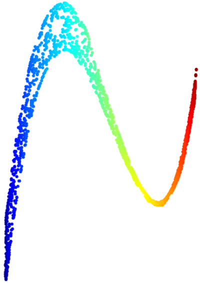
Semi-group error (SGE)

$$SGE = \| \quad K^{(2t)} \quad - \quad K^{(t)} \cdot K^{(t)} \quad \|$$

$$\left[\quad e^{-\frac{\|\cdot\|^2}{2t}} \quad \right] \quad \left[\quad e^{-\frac{\|\cdot\|^2}{t}} \quad \right]^2$$

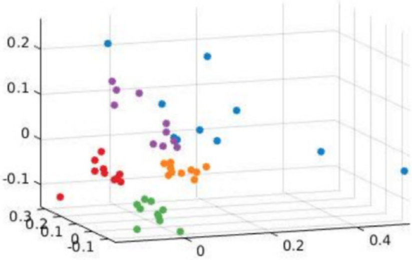
The right t should "minimize" SGE.





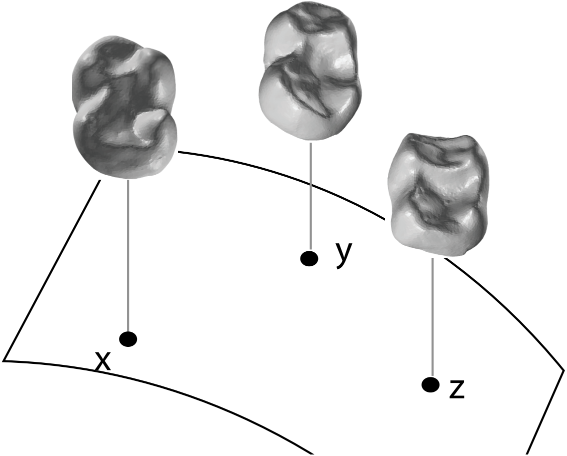
Visualize shape space of 5 species

Apply diffusion maps with semi-group test to 50 teeth



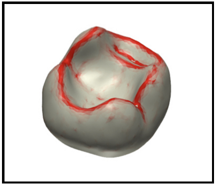
Color indicates different species

One step further: fibre bundle

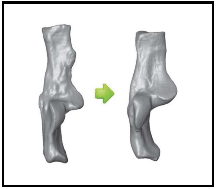


New statistics on fibre bundle help with study how shapes evolve.

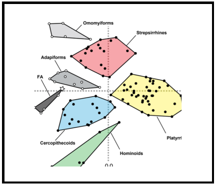
Math in the study of evolution



feature



shape distance



shape space

Key stat/math observations

PCA for computing curvature
Permutation matrix and shape distance
Semi-group properties for parameter tuning

Thank you for your accompany,
and happy holidays!

Shan Shan
sshan@math.duke.edu
http://sshanshans.github.com