TOWARDS CILIARY MOTION SUBTYPING: REPRESENTING PATIENTS AS MIXTURE OF MOTION PATTERNS

by

ALEKHYA CHENNUPATI

(Under the Direction of Shannon Quinn)

ABSTRACT

Cilia are microscopic hair like projections that lie on almost every cell of the body. Motile cilia have a rhythmic beating motion to clear mucus and irritants. If the mucociliary defense mechanism does not work properly, it leads to a wide spectrum of diseases called ciliopathies. Identifying ciliopathies early and implementing proactive therapies is clinically compelling to minimize procedural invasiveness. However, previous work in this area was limited to separating normal from abnormal ciliary motion, and ignored the existence of broader spectrum of ciliary beat patterns that may have clinical implications with different disorders. Hence, defining a universal, quantitative "language" that describes phenotypes of ciliary motion is of particular clinical and translational interest. The analysis presented here groups patients with similar ciliary motion patterns, establishing a platform that can unravel ciliary motion subtypes in patients.

INDEX WORDS: Cilia, ciliary motion subtype, dynamic textures, bag of dynamical systems, transformation invariant metrics, time series analysis.

TOWARDS CILIARY MOTION SUBTYPING: REPRESENTING PATIENTS AS MIXTURE OF MOTION PATTERNS

by

ALEKHYA CHENNUPATI

B. Tech, Jawaharlal Nehru Technological University, India 2011

A Thesis Submitted to the Graduate Faculty of the University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2017

TOWARDS CILIARY MOTION SUBTYPING: REPRESENTING PATIENTS AS MIXTURE OF MOTION PATTERNS

by

ALEKHYA CHENNUPATI

Major Professor: Shannon Quinn

Committee: Lakshmish Ramaswamy

Tianming Liu

Electronic Version Approved:

Suzanne Barbour

Dean of the Graduate School

The University of Georgia

December 2017

DEDICATION

I would like to dedicate this thesis to my husband, parents and all my family members for their constant support and encouragement.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and appreciation to Dr. Shannon Quinn for his infinite support, guidance and encouragement during the entire research. I would thank him twice for giving me an opportunity to serve as research assistant under him.

I would like to thank Dr. Lakshmish Ramaswamy for his continuous guidance, mentorship and valuable suggestions during the journey of my Masters. I would also thank Dr. Tianming Liu for his time and priceless suggestions.

I would like to appreciate my friends BahaaEddin AlAila and Ankita Joshi, MS students at University of Georgia for their input and constant support. Finally, I want to thank all my professors, Department of Computer Science and friends for making my stay at University of Georgia a pleasant and memorable one.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

1.1 Introduction

Cilia are microscopic hair like structures that extend from the surface of cells. They are found on cells in the lungs, kidneys, eyes, ear, nose, and brain [1-3]. Cilia play a vital role in human and animal development [3], but also play an important role in health upkeep and maintenance. Motile cilia lining in the nasal and lungs beat in a rhythmic motion to keep the airways clear of mucus and irritants, enabling easy breathing and exchange of oxygen and carbon dioxide [4]. If the cilia function is disrupted, it can result in a wide spectrum of diseases known as ciliopathy [6]. Primary ciliary dyskinesia (PCD) is an example of ciliopathy caused by a rare genetic disorder. The symptoms of PCD are malformed cilia ultrastructure that causes abnormal motion leading to ineffective mucociliary clearance [7].

Patients with these diseases suffer from frequent ear infections, chronic nasal congestion, male infertility [8] and respiratory infections. These infections can cause increased inflammation and scarring, ultimately resulting in bronchiectasis [9] and requiring organ transplantation [9]. Ciliary motion (CM) defects have been associated with increased respiratory complications and poor postsurgical outcomes [10-13]. Early diagnosis of CM abnormalities helps the clinician to institute respiratory therapies that could benefit patients without resorting to invasive, risky, and expensive surgeries.

Currently the most robust procedure to identify and diagnose ciliopathies is visual examination of the video microscope of nasal or bronchial biopsies by the clinicians for ciliary beat abnormalities. However, this procedure has several disadvantages. Relying on visual examination by expert reviewers is highly subjective, time consuming, and error prone [20-23]. Furthermore, these manual evaluations are not amenable to cross-institutional comparisons.

In Quinn et.al 2015, [24] the authors developed a computational framework that classifies normal and abnormal CM, achieving almost 93% accuracy. But, their work was limited to binary classification of normal and abnormal classification and were incapable of recognizing CM subtypes or novel categories of CM outside the binary paradigm. Given the variety of conditions associated with CM abnormalities, it is widely accepted by clinicians that there are more than two motion subtypes [30]. These subtypes may be diagnostically relevant to various disorders associated with certain ciliopathies. But, all the clinicians do not arrive at a consensus on the types of CM that exist.

Identifying CM subtypes is a very hard problem and requires unsupervised machine learning techniques to discover the latent motion patterns. Our work in this paper is an initial attempt to quantize CM and develop features amenable to identifying distinct CM patterns in patients. Our work draws from dynamic texture (DT) analysis in computer vision to extract features from raw CM, investigates several distance metrics for comparing geodesic DT representations of CM patterns, and explores the resolution of the resulting models [31]. In this way, we can recognize subtle variations in CM phenotypes with the ultimate goal of conducting more focused CM disease association studies.

1.2 Major Contributions

We treated CM as an instance of DTs which collectively form equivalence classes [25, 30], in which parameters describing the motion of two systems can be different in absolute terms but drawn from identical distributions. CM patterns inhabit a high dimensional geodesic space that cannot be analyzed using traditional linear (Euclidean) distance metrics. We tested several nonlinear distance metrics amenable to time series data: Bhattacharya distance [36], Martin distance [39, 40], Cepstral distance [36], and Kullback Leibler (KL) divergence [36]. Using these distance measures, we computed pairwise distances between DT representations of regions of interest (ROI) [24]. This pairwise distance formed a kernel matrix, which we used for further analysis. We applied clustering techniques like spectral and agglomerative clustering on these kernel matrices to observe the relationship between CM patterns for different patients. Due to the large number of ROIs, we chose t- distributed stochastic neighbor embedding (t-SNE) [64] for reducing the high dimensional data and visualizing the resulting CM-clustered patient data. t-SNE is proven to be a better technique for revealing the structure of high dimensional data that lies on several different but related low dimensional manifolds, such as image data [37].

Also, we applied few supervised techniques like support vector machine (SVM), random forest (RF), and k- nearest neighbors (KNN) to assess the validity of our DT-based features in relation to the manual labels supplied by clinicians. In this paper, we represent patients as a mixture of motion patterns to discover the CM subtypes.

We refined the process of selecting regions of interest by considering only the pixel intensities that vary a lot over time per each ROI. We assumed that the pixels that show large variation in pixel intensities over time are probably cilia that move in the videos and the pixels with least

pixel intensity variation over time are background of the video or cell bodies. We modeled the motion of cilia as a bag of dynamical systems [35]. The bag of systems (BOS) is similar to the bag of features (BOF) approach for document representation. Since CM patterns do not live in Euclidean space we applied non-Euclidean distance metrics as explained in the previous paragraph. But, now we obtained the kernel matrix on the pixel data instead of directly on ROI data.

Hence, we used nonlinear dimensionality reduction techniques to convert the high dimensional nonlinear data in to low dimensional Euclidean space which preserves the relationship in high dimensional nonlinear space. Then we applied a strategy similar to the bag of words approach in which we consider patients as documents and dynamical systems as the code words [35]. Then we applied clustering algorithms to represent the patients as mixture of CM patterns.

Chapter 3 provides a detailed explanation of the procedure we followed in identifying the CM subtypes and results analysis is explained in chapter 5.

CHAPTER 2

BACKGROUND AND RELATED WORK

In this chapter, we discuss in detail the structure of cilia and previous work done in this area.

2.1 Cilia: Structure and function

Cilia are microscopic hair-like organelles that extend from the surface of nearly all mammal cells [41]. They are found in linings of the airway, reproductive system, and other organs and tissues [1-3]. The length of a single cilium ranges from 1 to 10 micrometers [42]. In the airways, cilia function in concert with airway mucus to mediate the critical function of mucociliary clearance, cleansing the airways of inhaled particles and pathogens [4]. They also help in propelling sperm [1,4,7]. Cilia can be considered as a sensory cellular antenna that coordinate many cellular signaling pathways, coupling the signaling to ciliary motility or alternatively to cell division and differentiation [2]. Figure 1 shows the example of cilia in lungs [43].

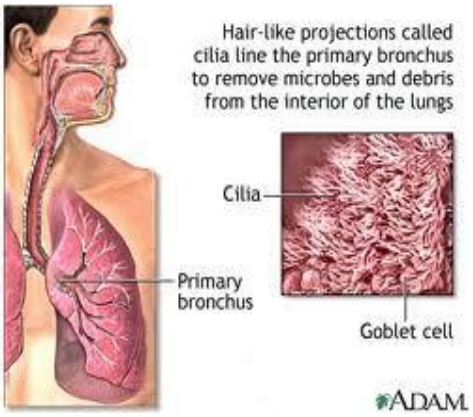

Figure 1: Example of cilia lying in lungs (Taken from [43])

2.2 Ciliopathies

If the proper motion of cilia is disrupted, it can lead to a wide spectrum of diseases collectively known as ciliopathies.

PCD is an example of a ciliopathy. It is a sinopulmonary disease arising from CM defects in the cilia of respiratory epithelia [44, 45, 46]. PCD is a genetic autosomal recessive disorder [7, 48] with incidence rate of about 1:15000 among Caucasians. Most infants with PCD experience breathing problems at birth, which suggests that cilia play an important role in clearing fetal fluid from the lungs [47]. The affected individuals develop frequent respiratory tract infections [8], as bacteria remain in the respiratory tract. People with PCD also have chronic nasal congestion and cough [48], which can develop into bronchiectasis [48]. This condition damages the airway passages leading from the windpipe to the lungs and can cause life-threatening respiratory problems. PCD in males lead to infertility [48, 49]. Infertility occurs in some affected females and is likely due to abnormal cilia in the fallopian tubes. Because motile cilia are also required for left- right patterning, PCD patients can exhibit mirror symmetric organ placement such as Kartagener syndrome or randomized left-right organ placement such as heterotaxy [50, 51]. Heterotaxy syndrome results from problems establishing the left and right sides of the body during embryonic development. Patients with congenital heart disease (CHD) and heterotaxy exhibit a high prevalence of CM defects similar to those seen with PCD [60]. This was associated with increased respiratory complications and poor postsurgical outcomes [11, 12, 60]. Similar findings were observed in patients with a variety of other CHD, including transposition of the great arteries (TGA), a CHD that may also arise from left-right patterning defects [61]. Diagnosing patients with CM abnormalities prior to surgery may provide a clinician with

opportunities to institute prophylactic respiratory therapies to prevent these complications. Access to early diagnosis and effective treatment is essential to curtail disease progression and to alleviate the burden of symptoms [67]. Hence there is a high need for early diagnosis of CM abnormalities.

2.3 Current ciliopathy diagnostic methods

Current procedures for diagnosing ciliopathies and associated disorders entail an ensemble of methods. One of these methods is to measure nasal nitric oxide (nNO) levels [52, 53]. Nasal nitric oxide levels have been demonstrated to correlate with certain ciliopathies [52]. But this is an unreliable technique amongst young children, and low nNO is also seen in other pulmonary diseases [53], hence this test leads to a high false positive rate. Another technique is to compute the ciliary beat frequency (CBF). Healthy CM tends to beat at higher frequencies (10-12 Hz) while abnormal motion is observed outside that range. However, there are numerous cases where these ranges are not definitive [14, 19-22]. Transmission electron microscopy (TEM) of ciliated epithelia can be used to find defects in the ciliary ultrastructure and determine ciliary orientation within cells [54,55], but few defects are very hard to detect due to low contrast [56] of the electron microscope. To identify such subtle ultra-structural defects a high-resolution immunofluorescence is extremely useful [57]. However, it needs the cells to be fixed, making the live observations impossible. Furthermore, CM defects can be observed with no detectable ultrastructural defects [58]. High-speed video microscopy is one of the most popular methods, both for its relative simplicity and for its ability to capture specific motion patterns. In practice, it has become an essential first step in the diagnostic evaluation of PCD and other ciliopathies [59].

Clinicians perform manual beat pattern analysis where a nasal brush biopsy from patient is collected, grown in a petri dish, and observed with a high speed digital microscope. The clinician observes the CM to identify motion defects. Although this is a widely-followed practice, reliance on the visual evaluations by expert reviewers makes these assessments highly subjective, time-consuming and error prone. It also heavily depends on how the clinician is trained in identifying such defects; without a universal CM language, each clinician is left to describe CM independently. These manual evaluations are not amenable to cross- institutional comparisons or any study of sufficient scale and quantity to draw definitive conclusions, necessitating quantitative identification of CM subtypes without manual intervention.

2.4 Related work

Given the deficiencies of the current diagnostic practices followed by clinicians, some work has been proposed to provide a quantitative definition of CM. Quinn et. al 2015 [24] proposed a computational framework to quantitatively assess the CM using DT analysis. DTs are sequences of images of a moving scene which exhibit temporal regularity subjected to stochastic noise [25-28]. Some examples of DTs include rippling water, flickering flames, and grass in the wind; each has a regular visual pattern that is combined with some stochastic behavior. CM is well-described as an instance of DT, since the motion of cilia is rhythmic but also subjected to stochastic noise that collectively determines the beat pattern. This assumption showed good results in interpreting CM in the case of binary classification. Therefore, in this paper we build on this work by also considering CM as an instance of DT.

2.5 Quantitative representation of CM

We quantitatively represent CM in the form of a DT. DT analysis relies on using a linear dynamical system (LDS) such as autoregressive (AR) models [38] to parameterize the components of DT motion. Instead of working directly on the grayscale pixel intensities, we used elemental components to quantitatively represent CM. Grayscale pixel intensities are not invariant to affine transformations, complicating the derivation of higher-order statistics such as AR models from the grayscale pixel values.

Hence, we used the elemental components obtained after computing the optical flow [29] of the grayscale CM. Using the optical flow [29] we can model the apparent change in CM between two consecutive frames as a vector field. This indicates the direction and magnitude of apparent motion at each pixel position in the ROI.

The following sections explains the related work done in the area, a procedure followed to quantitatively describe CM, and few limitations to the previous work.

2.6 Optical flow

The apparent motion of objects in a scene caused by the relative motion between an observer and the subject is called optical flow [29]. Optical flow represents the relative displacement of motion between a pair of consecutive frames from the original video. We do not explicitly track individual cilia when determining the CM, but rather estimate CM using the spatial and temporal derivatives of the optical flow. Using these derivatives of the optical flow we computed the elemental components of CM.

The size of the neighborhood and strength of contribution of the neighborhood to the final optical flow of the pixel of interest varies across implementations. In general, these constraints are

pooled over a neighborhood of pixels to smooth the effects of noise [29]. In [24], few techniques such as temporal smoothing, median filtering, and a few noise filtering techniques like spatial median filtering of dominant frequencies and principal component analysis (PCA) on AR models are used to remove lower modes of motion and keep only the dominant ones. The authors also applied Gaussian filtering with a scaling parameter while computing the optical flow and its derivatives. In our work, we used the Farneback optical flow algorithm, as implemented in OpenCV [62]. Farneback is a dense optical flow algorithm and computes optical flow for all pixels in the frame.

The main assumption of optical flow is that the intensities of an object does not change between consecutive frames (brightness constancy) and the neighborhood pixels have similar motion.

$$I(x, y, t) = I(x + u\delta t, y + v\delta t, t + \delta t) \quad (1)$$

Equation (1) indicates that the image intensity at a location *(x, y)* at time t $I(x, y, t)$ is preserved locally for small changes *(uδt, vδt)* in the next frame taken after δt time. Here *(u, v)* are the horizontal and vertical image velocity components of the optical flow vector $\overrightarrow{f^T}$ at pixel location *(x, y)*. Applying Taylor series approximation on the right-hand side, removing the common terms, and dividing by *dt*, we get the following equation

$$I_x u + I_y v + I_t = 0 \text{ where} \quad (2)$$

$$I_x = \frac{\delta I}{\delta x} \qquad I_y = \frac{\delta f}{\delta y}$$

$$u = \frac{\delta x}{\delta t} \qquad v = \frac{\delta y}{\delta t}$$

In Equation (2), I_x and I_y are components of the spatial gradient, and I_t is the temporal gradient. The optical flow vector (u, v) is estimated from an overdetermined system of linear equations

which are formed from the gradient constraint that is pooled over small image neighborhood around pixel (x, y).

The optical flow vector $\vec{f} = (u, v)^T$ provides information on the image dynamics; the first-order flow derivatives (u_x, u_y, v_x, v_y), can be derived from the optical flow to represent an affine model of optical flow. For derivation of the optical flow and for computing its derivatives to obtain the final elemental components, please refer to the Appendix.

Fig 2 illustrates the process of computing optical flow and elemental components of 3 frames in a CM video. The elemental components of the CM are obtained by spatial and temporal derivatives of the optical flow. From the optical flow, we compute three elemental components: rotation, divergence, and deformation.

Rotation indicates the curl of cilia, in radians per second. Divergence captures scaling or motion of the cilia towards and away from us (zoom in and zoom out). For 2D videos where most CM lies within the plane of the camera, divergence is not a useful quantity. Deformation is the biaxial shear of the cilia (crushing and squeezing). Elemental components are computed at each pixel, and therefore have the same dimensionality as the original grayscale data. However, it should be noted that deformation is a vectorial quantity and consists of horizontal and vertical components (similar to optical flow), whereas rotation is a scalar quantity (similar to grayscale).

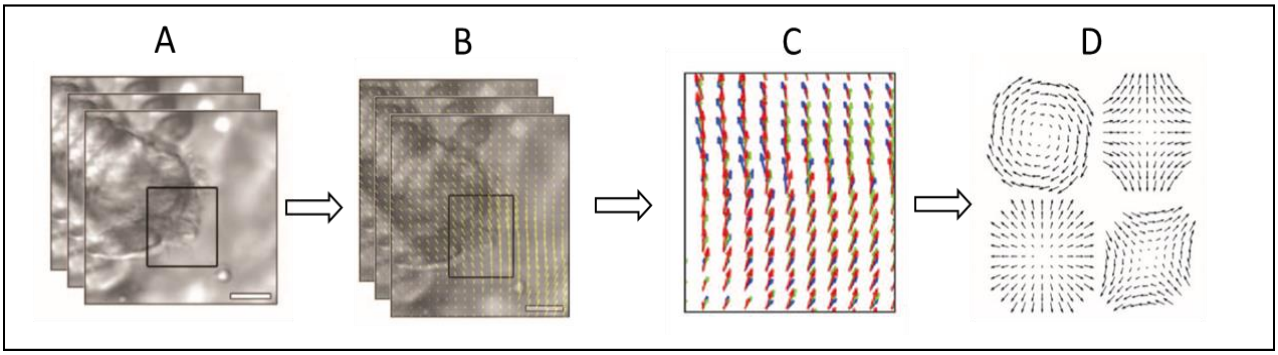

Figure 2: Compute optical flow for three frames. (A) Stacked frames indicating the cilia and cilia body from a video of nasal brush biopsy Black box is the ROI selected. (B) Shows the direction and magnitude of optical flow for each pair of frames in the video. (C) Changes in the optical flow to compute elemental components. Red arrows indicate optical flow at time t, green arrows show optical flow at frame t+1, blue arrows are optical flow at t+2. (D) Elemental components of optical flow, top left is rotation which we used in this paper and bottom left is divergence. Top and bottom right are deformation templates. Deformation is vector quantity.

Previous studies [24] show that rotation and deformation components for healthy CM displayed strong periodic behavior and a wide range of magnitudes. In contrast, abnormal cilia showed weak periodic behavior in addition to magnitude variations heavily centered around 0, indicating very little movement overall. Between the two elemental components, rotation could accurately differentiate the abnormal CM [24]. Clinical studies describe the abnormal CM as having reduced beat amplitude, stiff beat pattern, and failure to bend along the length of the ciliary shaft [19, 20]. This observed stiffness is shown most apparently with rotation elemental component [24], suggesting a translational pathway that rotation quantity captures a biologically meaningful phenotype. Therefore, we used the rotation component in our analysis.

2.7 Auto regressive models

AR models are the state of art DT analysis methods that are useful in representing periodic signals containing noise [25, 31-33]. Linear models are sufficient to capture the motion between successive frames [24]. We represent the dynamics of CM using two equations:

$$\vec{y}_t = C\vec{x}_t + \vec{u}_t \quad (3)$$

$$\vec{x}_t = A_1\vec{x}_{t-1} + A_2\vec{x}_{t-2} + \cdots + A_d\vec{x}_{t-d} + \vec{v}_t \quad (4)$$

where Eq. 3 models the appearance of the cilia $\vec{y}$ at a given time *t* plus a noise term $\vec{u}_t$. Eq.4 represents the state $\vec{x}$ of the CM in a low-dimensional subspace defined by an orthogonal basis *C* at time *t*, plus a noise term $\vec{v}_t$, and how the state changes from *t* to *t* + 1.

Eq. 3 is a decomposition of each frame of a CM video $\vec{y}_t$ into a low-dimensional state vector $\vec{x}_t$ using an orthogonal basis *C*. Singular Value Decomposition (SVD) was used to derive this basis. The input to the SVD is raster-scan of the original video; the video is restructured into a matrix where each row corresponded to a single pixel from the video and each column is a frame (or the value of that pixel in a given frame). Therefore, if the height and width of the video in pixels were given by *h* and *w* respectively and the number of frames as *f*, the dimensions of the raster-scanned matrix would be $hw \times f$.

The main assumption in DT analysis is that the DT lives in a low-dimensional subspace as defined by the principal components *C*; a significant majority of the variance in
the data can be explained with only a few dimensions [24]. Once the data $\vec{y}_t$ is projected in to this subspace, we can model the motion of the DT $\vec{x}_t$ using a relatively few parameters by virtue of its low dimensionality relative to $\vec{y}_t$. This motion can be described as a linear process:

at time *t+1*, the position of cilia in this low dimensional space is a linear function of its position at some number *d* of previous positions. This intuition is represented in Eq. 4: CM at position $\vec{x}_t$ is a function of the sum of *d* of its previous positions $\vec{x}_{t-1}$, $\vec{x}_{t-2}$, $\vec{x}_{t-d}$, each multiplied by its corresponding coefficients A = {A_1, A_2, ..., A_d}. The noise terms $\vec{u}$ and $\vec{v}$ are used to represent the residual difference between the observed data and the solutions to the linear equations; this is modeled as Gaussian white noise.

Each DT is represented as a combination of its coefficients *A* and its subspace *C* when comparing the DT using AR models. In other words, a DT model *M* is represented using its parameters *(A, C)* [35]. However, our work differs from most other DT analyses that attempt to differentiate distinct instances of DTs. Here, we hypothesize that all CM lives within same subspace *C*, shared by all the instances of CM [24]. What differs, we claim, is the movement of the cilia within this subspace. This movement is captured solely by the coefficients *A*; therefore, we represent each instance of CM using only the coefficients.

The orientation-invariance property of the elemental components enables our use of PCA to define the low-dimensional orthogonal basis *C*. PCA realigns the axes of the data in the directions of maximal variance. Performing PCA on a video of raw pixel intensities would result in different principal components depending on the relative orientations of the structures in the video. For example, if a video depicting a profile-view of cilia beating from left to right, then principal components will be different from those after rotating it 90 degrees. However, since rotation and deformation are computed from the magnitudes of optical flow derivatives, the relative orientation of structures defined by the pixel intensities does not matter in the

computations. This makes rotation and deformation orientation-invariant and allows CM to be compared irrespective of relative ciliary orientation across multiple videos.

2.8 Previous approach and few limitations in it

In [65], the authors built automated image analysis methods to eliminate the manual examination of beat patterns. While this research served as a basis for our analysis to use some of their techniques, there is a need to quantitatively identify various CM subtypes. Our work helps in achieving this goal.

In the previous approach of [24], Quinn et. al 2015 successfully quantified CM data and were able to identify normal and abnormal CM with almost 93% accuracy. But grouping of CM into one of the binary types is an oversimplification that ignores the underlying spectrum of CM subtypes.

CHAPTER 3

CILIARY MOTION SUBTYPING AND OUR APPROACH

This chapter justifies the need for CM subtyping, provides the motivation for our work and explains the techniques used.

3.1 Why CM subtyping is needed?

The previous approach [24] was successfully able to discriminate normal and abnormal patterns, ignoring the continuum of ciliary beat patterns. This continuum most likely has biological and clinical implications with various disorders and ciliopathies. This makes the identification of CM subtypes and their association with these disorders of translational interest. One of the main roadblocks in identifying distinct CM subtypes is that no consensus exists as to the exact number of subtypes or how to objectively define them. This work focuses on identifying the latent CM patterns in the patient that can later help to quantify the CM subtypes computationally. Using the computational pipeline outlined here, we conclude that at least 10 CM subtypes exist. We also compare the DT representations of CM developed in our analysis with one expert categorization of CM into 4 beat pattern types and discuss the results of this comparison.

3.2 Different ciliary subtypes

There are multiple phenotypes associated with the CM, though an exact number is still a scientific research area. But all the clinicians in Children's hospital of Pittsburgh and Children's National Medical Centre agree upon that there are at least 4 subtypes. Hence here in fig 3 we

present the 4 subtypes, hand drawn for easier identification. But this does not mean that there are just four CM types that are accepted by all the clinicians. There are several other motion patterns.

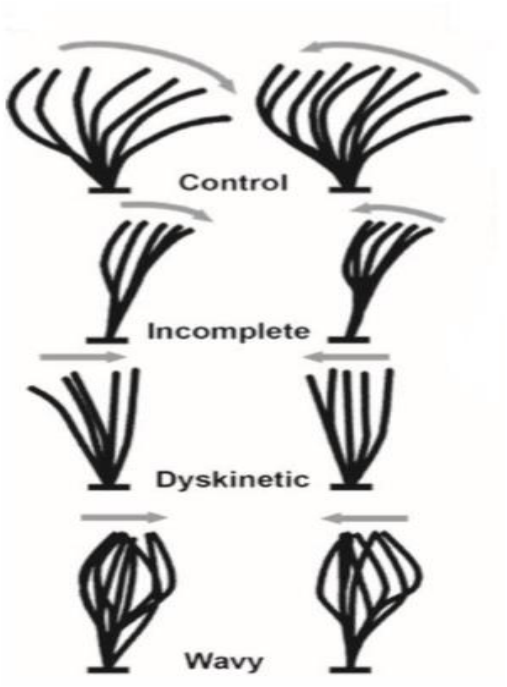

Figure 3: Hand drawn diagrams of CM subtypes. Although there are multiple subtypes, these 4 are the widely-accepted ones. (Taken from 24)

3.3 Our approach

CM represented in digital videos inhabit a high dimensional space (since all video data is high dimensional) although we assumed they exist on a common low-dimensional manifold. The space is geodesic and highly nonlinear; thus, linear distance metrics like Euclidean cannot be used to compare motion patterns. In addition, we also need metrics that are transformation invariant. The following explains the need for transformation invariant metrics and the metrics we chose.

3.3.1 Transformation invariant metrics

To quantitatively differentiate the CM between ciliary beat patterns represented as AR processes, let us take two instances of CM $\theta_1 = (C, A, Q)$ and $\theta_2 = (C^1, A^1, Q^1)$. These parameters constitute the principal components C, the state space coefficients A, and the covariance matrix Q for the residual noise vector $\vec{v}_t$. Since we assumed that cilia lie in common state space C, we consider $C = C^1$ for all C^1. We are left with coefficients A and the covariance matrix Q to distinguish between CM phenotypes. However, it is not sufficient to check for absolute equality of parameters as instances of CM (and any DT represented as an AR process) may be generated by an equivalence class of models. This means, for any invertible square matrix M, the model $(CM^{-1}, MAM^{-1}, MQM^{T})$ generates a sequence of frames from the same distribution as (C, A, Q) [36]. Thus, any metric for comparing instances of CM as represented using AR models must be transformation-invariant. Here, we experimented with four metrics that capture this property.

3.3.2 Time series spectrum

We measure the divergence between the DT models in terms of the Fourier transform of the autocovariance of the time series or its spectral density [63, Ch3].

The spectrum $F(V_k)$ is estimated using the fast Fourier transform (FFT) of the raw time series. Let X be the state space projection of the original data of dimensions $q*M$, q is the number of principal components used, and M is the number of features.

FFT of the series, f_k is calculated by first computing the component wise FFT

$f(i,:) = fft(X(i,:))$ and then set $f_k = f(:,k)$.

The periodogram G_k is computed using $G_k = f_k f_k^*$.

The spectrum is given by smoothing G with a window of size 2H + 1, yielding

$$F(V_k) = \sum_{i=k-H}^{k+H} G_k \quad (5)$$

This is called the time series method.

3.3.3 Distance between DTs

Now that we calculated the spectrum, we defined the distance between DTs using KL divergence, Bhattacharya distance, Cepstral distance, and Martin distance.

3.3.3.1 KL divergence

To compute the KL divergence of our DT model, suppose we have two videos $(C_j, A_j, Q_j)_{j=1,2}$, the spectral densities $F_j(v_k)$ for both the videos are computed using equation 5. From this definition, the KL distance from (C_1, A_1, Q_1) to (C_2, A_2, Q_2) is

$$D_{KL}(F1, F2) = \sum_{0<vk<\frac{1}{2}} [trace\{F_1(V_k)\, {F_2}^{-1}(V_k)\} - \ln \frac{|F_1(V_k)|}{|F_2(V_k)|} - N] \quad (6)$$

However, its lack of symmetry presents challenges for defining a cohesive and intuitive space of CMs.

3.3.3.2 Bhattacharya symmetric divergence

As KL divergence is non-symmetric, we therefore included the Bhattacharyya symmetric divergence which measures the dissimilarity between the distributions of dynamic features. We represent this distance D_B between (C_1, A_1, Q_1) to (C_2, A_2, Q_2) after computing spectral densities $F(V_k)$ from equation (5) using

$$D_B(\alpha, F1, F2) = \frac{1}{2}\sum_{0<vk<\frac{1}{2}}[\ln\frac{|\alpha F_1(V_k)+(1-\alpha)F_2(V_k)|}{|F_2(V_k)|} - \alpha\ln\frac{|F_1(V_k)|}{|F_2(V_k)|}], \quad (7)$$

where $0 < \alpha < 1$ is a tuning parameter

[36] shows that the success rate did not depend sensitively on α near the middle of the interval (0,1). Thus, we took $\alpha = 0.5$. This metric satisfies the triangle inequality and obeys all metric axioms.

3.3.4 Cepstral distance

The cepstrum of a time series can be derived from the frequency domain representation in the same way that the time series comes from the time domain. Intuitively, peaks in the cepstrum correspond to "echoes" in the signal. The cepstrum coefficients are powerful features for characterizing speech and music signals [34].

To compute the Cepstral distance, we applied the discrete Fourier transformation (DFT) to each video patch. The cepstrum of a multivariate time series $(x_t)_{t=1}^{T}$ is the inverse DFT of the logarithm of the DFT of (x_t):

$$(\hat{x_t})_{t=1}^{T} = IDFT(\ln(DFT((x_t)_{t=1}^{T})) \quad (8)$$

Here the DFT of a sequence of vectors is taken component wise. Thus, the cepstral coefficients of a multivariate time series are vectors. The cepstral distance is defined as

$$\sum_{t=1}^{n} ||\widehat{x_t} - \widehat{y_t}|| \quad (9)$$

3.3.5 Martin distance

Martin distance was used for discriminating between the parameters of LDS. It is defined over the subspace angles between two systems. Martin distance over subspace of the system C and motion parameter A is defined as

$$A^T P A = -C^T C \quad (10)$$

where, for 2 patches v_i and v_j with q subspace dimensions, and patch size w,

$$P = \begin{bmatrix} P11 & P12 \\ P21 & P22 \end{bmatrix} \in R^{2q\ X\ 2q} \quad (11)$$

$$A = \begin{bmatrix} A_{v_i} & 0 \\ 0 & A_{v_j} \end{bmatrix} \epsilon R^{2q\ X\ 2q} \quad (12)$$

$$C = [C\ C] \in R^{w^2\ X\ 2q} \quad (13)$$

Once we find P by solving the above Lyapunov equations, we get a symmetric matrix of the components P and perform the following eigen decomposition. Each eigenvalue λ_i of this matrix is the cosine of the subspace angle θ_i squared.

$$\text{Cos}^2\theta_k = \text{k}^{\text{th}} \text{ eigenvalue } (P_{11}^{-1} P_{12} P_{22}^{-1} P_{21}) \quad (14)$$

Now that we have values of this matrix, we can compute Martin distance D_M between video patches V_i and V_j using

$$d_M(v_i, v_j)^2 = ln \prod_{k=1}^{q} cos^2 \theta_k \quad (15)$$

We used martin distance for 3rd order AR processes. The main assumption of martin distance is that it depends on the subspace angles. But we used pixel data for every ROI which is a 1-D space, which has no subspace for cilia. *C* is essentially an identity matrix, for which calculating subspace angles does not make sense. While we initially tried Martin distance, we did not obtain meaningful results; thus, we will not discuss them further.

Once we computed the pairwise kernel using above 3 distance metrics, we get a square matrix of dimensions – (number of ROI*top pixel intensities) x (number of ROI * top pixel intensities).

3.4 Bag of dynamical systems approach

This approach models each patient as a collection of LDSs describing the dynamics of spatiotemporal video patches. This BOS representation is analogous to the BOF representation of document analysis. In this case, we use LDSs as feature descriptors and essentially consider each patient to be analogous to a single "document". This poses several technical challenges to the BOF framework. Most notably, LDSs do not live in a Euclidean space, hence novel methods for clustering LDSs and for computing code words of LDSs are used. Our framework makes use of nonlinear dimensionality reduction and clustering techniques combined with KL divergence, Bhattacharya distance, cepstral distance for LDSs for tackling these issues.

The typical steps followed in the BOF framework are: (1) Features and their corresponding descriptors are extracted from all the images in the training set, (2) A codebook is formed using clustering methods such as k-means, where the cluster centers represent code words in the

codebook, (3) Each image in the training dataset is represented using the codebook, and (4) A classifier is chosen to compare a new query image to the training set and thus infer its category. We describe the corresponding BOS analogue to the BOF steps. As we use the parameters of a LDS as feature descriptors, we now have features in a non-Euclidean space. In the traditional BOF framework [25], once we extract feature points and their corresponding descriptors, the descriptors are clustered using an algorithm such as k-means to form the codebook. But, in our case the descriptors are parameters of a LDS, i.e. pixel data which lie on non-Euclidean space. We cannot apply clustering algorithms that assume a Euclidean space. But if we can find a low-dimensional Euclidean embedding for these points, we can apply clustering techniques. The following section explains the process of clustering and the code book formation.

3.5 Codebook formation

We applied a variety of nonlinear dimensionality reduction techniques using Laplacian Eigen Maps (LEM), Multi-Dimensional Scaling (MDS), Isomap, and PCA. These techniques work with pairwise distances between points in a high dimensional space. In this paper, we made use of the fact that LDS is established with several distances (KL divergence, Bhattacharya distance, and cepstral distance) and used the corresponding distance metric to perform dimensionality reduction and clustering to form the code book.

Given the set of features $\{M_i\}_{i=1}^{T}$, where T represents the total number of features extracted from the videos in the training set and we have N such data point, we first form the matrix $D \epsilon R^{N\,X\,N}$ such that

$$Dkl_{ij} = d_{kl}(M_i, M_j) \tag{16}$$

$$Dbc_{ij} = d_{bc}(M_i, M_j) \tag{17}$$

$$Dcep_{ij} = d_{cep}(M_i, M_j) \quad (18)$$

Once pairwise distances are available between the features, nonlinear dimensionality reduction techniques are used to obtain a low-dimensional embedding of these points $\{e_i \in R^{d_e}\}_{i=1}^{N}$ where $d_e$is the dimension of embedding. This low dimensional representation helps us to obtain a set of Euclidean points, which preserves the relationship in the high-dimensional nonlinear space. Now we applied clustering algorithms on $\{e_i\}_{i=1}^{N}$ since this low dimensional space is Euclidean.

After applying clustering algorithms, we get *k* cluster centers $\{k_i\}_{i=1}^{K}$. But, these cluster centers do not correspond to any of the original LDSs. Additionally, there is no explicit way to go from the lower dimensional embedding to the original space. Hence, to select the code words $\{F_i\}_{i=1}^{K}$, we chose the corresponding systems in the high dimensional space whose distance to the cluster center in the lower dimensional space is the least

$$F_i = KL_p$$

$$F_i = BC_p$$

$$F_i = Cep_p$$

where $p = \arg min_j^{||e_j - k_i||^2}$,

KL – KL divergence, *BC* – Bhattacharya distance and *Cep*- Cepstral distance.

In this way, we obtain our codebook $C = \{F_1, \ldots, F_k\}$, where $F_i = (M_i)$, where M is the original set of features. This essentially means that we mapped *k* low-dimensional cluster centers to the nearest original high-dimensional data points. During the query phase, each detected feature is associated with model parameters M. The membership to the code word is given by

$$membership = arg\ min_i\ d_{KL}(M, F_i) \quad (19)$$

3.6 Representing patients using codebook

Now that we have *k* code words available, we represent each patient as a mixture of these codewords, analogous to how documents are represented as a mixture of words in the bag-of-words model. This can be done using weight vector

$$W = \{w_1, w_2, w_3 \ldots . w_k\} \epsilon R^k \quad (20)$$

We used couple of approaches to represent this weight vector.

Let us assume that code word *k* occurs N_{ki} times in the i^{th} patient and there are total of N_i code words in the i^{th} patient. Term frequency (TF) is computed using

$$w_{ik} = \frac{N_{ki}}{N_i} \quad (21)$$

Let *V* be the total number of video sequences and V_i be the total number of patients in which code word *i* occurs. Term frequency inverse document frequency (*TF-IDF*) is defined as

$$w_{ik} = \frac{N_{ki}}{N_i} \ln(\frac{V}{V_i}) \quad (22)$$

Once the weight vector *W* is computed, it is normalized by L_1 norm to become a histogram. We applied L_1 norm only for TF, so that highly weighted code words do not completely dominate the least weighted code words.

To compare the weight vectors, we used the standard distances between histograms such as χ^2 distance and cosine similarity.

3.6.1 χ^2 distance

To compare the weight vectors between patients, the standard distances between the histograms, χ^2 distance is used. χ^2 distance is defined as

$$d_{\chi^2}(W_1, W_2) = \frac{1}{2}\sum_{i=1}^{K}\frac{|w_{1i} - w_{2i}|}{w_{1i}+w_{2i}} \quad (23)$$

We applied χ^2 distance for the normalized TF weight vector and applied cosine similarity for TF-IDF weight vector.

3.6.2 Cosine similarity

Cosine similarity computes the similarity as the L2- normalized dot product between two weight vectors. Cosine similarity between 2 weight vectors W_1 and W_2 is defined as

$$d_{cosine}(W_1, W_2) = \frac{W_1 W^T{}_2}{||W_1||||W_2||} \quad (24)$$

Since the Euclidean (L2) normalization projects the vectors on to the unit sphere, this is called cosine similarity and their dot product is then the cosine angle between the points denoted by the vectors.

After computing the kernel matrix using χ^2distance and cosine similarity we applied both classification and clustering algorithms.

3.7 Classification

For the given training data set $\{(W_i, l_i)\}_{i=1}^{V}$, where $l_i \, \epsilon \, \{1, \dots, m\}$ denotes class labels of the weight vector, our main aim is to find out the class label of a new weight vector W_t. To compare the training weight vectors and new unseen query weight vectors, the standard distances between the histograms, χ^2 distance and cosine similarity are used.

Classification is a supervised machine learning technique mainly useful for separating categorical data which forms a model using training examples. When new unseen data is given to

the model it tries to predict based on the learning experience and try to classify it into one of the labels. This method works only if we train the model using known labels. The main aim of classification is to generalize the unobserved data. There are multiple classification algorithms available and different algorithms implement different assumptions about the data. In this work, we experimented with three classification algorithms to see which works best on our data. Each patient is marked on a scale of 1 through 4 by our collaborating experts, where 1- corresponds to normal, 2- close to normal, 3- close to abnormal, and 4- abnormal. We trained the model using these labels and queried on new patients.

It is important to note that these labels are not an indicator of CM subtype, but rather a rudimentary measure of degree of CM abnormality. There is some relationship between the two but this relationship is neither necessary nor sufficient to identify CM subtypes. Nevertheless, we need some basis to compare our results against. So, we used these labels as one of the criteria. The accuracy of prediction does not yield information about CM subtypes, but serves to deepen our intuition for the data and evaluate our quantitative CM representations.

We trained each model using stratified k- fold cross-validation, which splits the data into training and testing folds while also preserving the percentage of samples of each class that are in each fold. We used 3-fold random cross validation and computed the mean accuracy of classification in each round.

Our first classifier was a *k*-nearest neighbors model, or KNN. Using the KNN model to predict the category of a query patient, the majority class label of the query patient's *k* closest weight

vectors from the training dataset is used to predict its label. k -NN is an instance based learning where the function is only approximated locally and all computations are deferred until the classification is applied. We used weighted k- NN where the weights assigned to the neighbors are proportional to the inverse of the distance from the query point. We used 3 nearest neighbors by prioritizing few weights based on the distance between them.

Our second classifier was the Support Vector Machine, or SVM. SVMs perform classification by defining a hyperplane that most accurately separates training data into its constituent classes. When trained, an SVM builds a maximum margin hyperplane to separate the classes; when an unseen example is given it tries to assign this to one of the categories. This means, given the training data, the algorithm outputs an optimal hyperplane which can categorize new examples. Here we used a multi-class generalization of the core SVM referred to as a Nu-support vector classification (NuSVC), which works like SVMs but uses a parameter to control the number of support vectors. We used our precomputed distance kernels with the upper bound fraction of training errors of 0.2. We trained our data using NuSVC with these parameters and tested on unseen patient data.

We also experimented with three ensemble classifiers. The first was Random Forest (RF), that operates by constructing multiple decision trees classifiers on various subsamples of the dataset and uses averaging to improve the prediction accuracy and to control overfitting of the data. In training the model when we split a node in a tree, the split we chose was not the best split among all the features but rather the best split among a random subset of features. Because of this randomness, the bias of the forest slightly increases (with respect to the bias of a single non-

random tree) but since we average it the variance also decreases usually more than compensating for the increase in bias. Hence, we get an overall better model.

The second ensemble classifier was bagging that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. Such a meta-estimator can typically be used to reduce the variance of a black-box estimator, by introducing randomization into its construction procedure and then making an ensemble out of it [69].

The third and the final ensemble classifier was ada boost to fit a sequence of weak learners on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction. The data modifications consist of applying weights w_1, w_2, ..., w_N to each of the training samples. Initially, those weights are all set to $w_i = 1/N$, so that the first step simply trains a weak learner on the original data. For each successive iteration, the sample weights are individually modified and the learning algorithm is reapplied to the reweighted data. At a given step, those training examples that were incorrectly predicted by the boosted model induced at the previous step have their weights increased, whereas the weights are decreased for those that were predicted correctly. As iterations proceed, examples that are difficult to predict receive ever-increasing influence. Each subsequent weak learner is thereby forced to concentrate on the examples that are missed by the previous ones in the sequence [70].

3.8 Clustering

Clustering is an example of unsupervised machine learning technique which describes the hidden structure from unlabeled data; in this case, our goal was to use clustering to group patients together who exhibited similar CM patterns. By applying clustering techniques, we can identify a structure to totally unstructured data. In our paper, we applied clustering algorithms on the kernel matrix of the weight vectors of the patients. This enables us to identify any hidden pattern in the CM of the patients. We applied spectral clustering algorithm to our data.

Spectral clustering relies on the eigen-decomposition of the pairwise similarity matrix of the data in order to reduce the dimensionality of the data and generalize anisotropic latent data distributions. In other words, it applies clustering to the principal components of the underlying connectivity graph of the data. It is very useful when the structure of the individual clusters is not Gaussian. Here we applied spectral clustering to our precomputed kernel matrices (kernel obtained after computing cosine similarity and χ^2 distance).

We also applied t-SNE, an algorithm which is well suited for embedding high dimensional data into 2 or 3 dimensions, which can then be useful for visualization on a scatter plot. T -SNE converts the similarities between data points to joint probabilities and tries to minimize the KL divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data. It has a non-convex cost function i.e. with different initializations we can get different results [64].

t- SNE focuses on the local structure of the data and tends to extract clustered local groups of samples. This ability to group samples based on the local structure might be beneficial to visually

disentangle a dataset that comprises several manifolds at once. Optimizing t-SNE depends on various factors like perplexity, learning rate, maximum number of iterations [66]. Perplexity is twice the Shannon entropy of the conditional probability distribution. Larger perplexities lead to more nearest neighbors and less sensitive to small structure. Larger datasets tend to require larger perplexities. The maximum number of iterations is usually high enough and does not need any tuning. The early exaggeration of the joint probabilities in the original space can be artificially increased by multiplying with a given factor. Larger factors result in larger gaps between natural clusters in the data. If the factor is too high, the KL divergence could increase during this phase. Usually it does not have to be tuned. A critical parameter is the learning rate. If it is too low gradient descent will get stuck in a bad local minimum. If it is too high the KL divergence will increase during optimization.

In our analysis, we reduced the number of dimensions to two, perplexity to thirty, early exaggeration to four, learning rate as thousand with thousand iterations, and precomputed distance metric. We obtained an embedding matrix and visualized it using a scatter plot of the embedding, where we color coded each patient based on the spectral clustering labels.

CHAPTER 4

ARCHITECTURE

This chapter explains the data used in our work and the architecture of our pipeline.

4.1 Data Used

We performed our analysis on the data provided by Children's Hospital of Pittsburg. All study protocols were approved by the University of Pittsburgh Institutional review board. This data cohort consisted of nasal brush biopsies of the patients suffering from PCD, few suffering from TGA, and some normal patients (people who do not have abnormal CM). Nasal epithelial tissue was collected by curettage for high speed digital video microscopy (using 200 Hz) using well established methods.

Video samples were examined by the clinicians at University of Pittsburgh and given a ground truth CM identification from 1(normal), 2(Probably normal), 3(probably abnormal) and 4(abnormal) that reflects the degree of normal and abnormal CM. These numbers are assigned by majority rule meaning that a group of clinicians observed these videos and marked them on a scale of 1 through 4. If all the clinicians agree that the patient is normal then patient is marked 1 and if all the clinicians agree that a patient is abnormal then the patient is marked as 4. If there is a disagreement among the clinicians about a patient, then they are marked as either 2 or 3.

In this paper, we experimented with biopsies recorded with high speed digital videos from 18 patients, with 46 ROI out of which 10 people are normal, 2 closer to normal, 2 of them are closer

to abnormal and 4 patients are abnormal. Table 1 represents the distribution of ROIs for each patient based on their marked labels.

Label marked for a patient	Number of ROIs selected
1	12
2	10
3	10
4	14
Total ROIs	46

Table 1: Represents the number of ROIs selected for patients labeled by the clinicians.

The ROIs were manually drawn by the clinicians where cilia exist. There were typically multiple ROIs per video. This helped in removing some of the noise, background, and unnecessary data for further computations. Once we extracted the ROI, we performed more preprocessing steps as explained in next section.

4.2 Architecture

ROIs from high speed digital video are drawn and the following steps are followed

1. Optical flow is computed in user specified ROI for the digital videos. This helps in indicating the direction and magnitude of the apparent motion at each pixel position in the ROI. The derivative of the optical flow- rotation component is used in our analysis instead of gray scale pixel intensities.

2. From the rotation data, we located a pixel in the middle of each ROI with dominant frequency and expanded a *w* x *w* (15 x 15) box called a "patch." We truncated each frame of the video at 250 frames and flattened the 15 x 15 patch in each frame to a single 225 length vector. We repeated this process for all 250 frames generating a data structure of size 225 X 250. We repeated this process for all ROIs, appending each patch to the end of previous one. Hence, for our data set with 46 ROIs, we ended with a data structure of size 225 x (250*46) i.e. a matrix of dimensions 225 x 11500.
3. From the 225 pixels in each patch, we picked top 100 pixels per frame with high standard deviation across frames assuming that the pixels with high standard deviation most likely represented cilia that was moving. We also cut down the number of frames to 200. From this, we reduced our data matrix to 100 x 200 dimensions per each ROI. We then stacked each such pixels across all the patients for all ROIs. For 46 ROIs, we ended up with a matrix of dimensions 4600 x 200.
4. We computed the pairwise kernel matrix for each of the distance metrics. We calculated distances between every pair of 4600 x 200 with 4600 x 200, which yielded a kernel matrix of dimensions 4600 x 4600.
5. We applied bag of dynamical system approach on this matrix, computing AR parameters for each pixel. We converted this high dimensional data to a low dimensional Euclidean space using PCA, LEM, MDS, and isomap encompassing a manifold of 4600 x 2. This low dimensional embedding gives us a set of Euclidean points which preserves the relationship in the high dimensional nonlinear space.
6. K-means clustering was applied and k clusteroids were obtained for the embedding. We used k = 1000, hence we got 1000 cluster centers corresponding to each embedding.

7. We found the corresponding systems in high dimensional space whose distance to cluster center in low dimensional space was the smallest. These indices with the corresponding points were matched against original data (4600 x 200). These were our code words, we have 1000 such code words (each vector of length 200).
8. Step 7 converts the unknown low dimensional embedded cluster centers into known corresponding clusteroids in high dimensional space. We then calculated the distance from every code word (1000 x 200) to the original data (top pixel data of dimensions 4600* 200) and considered the code word closest to each pixel as the membership to the code word. Here we used the corresponding distance metric to calculate the smallest distance. Hence, we now obtained a data structure with list of patient ids and its closest code word with 4600 such items in the list.
9. Each patient is represented using a code book. This was done using a weight vector of k (1000) code words $W = \{w_1, w_2, w_3 \ldots . w_k\} \epsilon R^k$. We represented it using TF and TF-IDF. TF is computed using $w_{ik} = \frac{N_{ki}}{N_i}$, where N_{ki} – Number of times code word *k* occurs in the i^{th} patient and N_i – Total number of code words in the i^{th} patient. We have 1000 such weight vectors for all 18 patients.
10. TF-IDF is computed using $w_{ik} = \frac{N_{ki}}{N_i} \ln(\frac{V}{V_i})$, where V is the total number of video sequences and V_i is the total number of patients in which code word *i* occurs. We have 1000 such TF- IDF weight vectors per each patient.
11. We normalized the TF weight vector using L_1 norm to become a histogram. We applied L_1 norm only for TF so that highly weighted code words do not completely dominate the least weighted code words.

12. Then we computed standard distances between histograms on both TF and TFIDF weight vectors using χ^2 distance and cosine similarity on 18 patients. We now get a kernel matrix of size 18 x 18.
13. We applied spectral clustering to represent the patients with similar CM pattern into same cluster. We visualized the results using t-SNE plot.
14. We ran our classifiers on the precomputed kernel matrix from step 12: SVC, KNN and random forest. The true labels based on the CM condition of the patient were marked by the clinicians. We trained and tested classifier using 3-fold cross validation.

Figure 4 explains the step by step process performed in our paper.

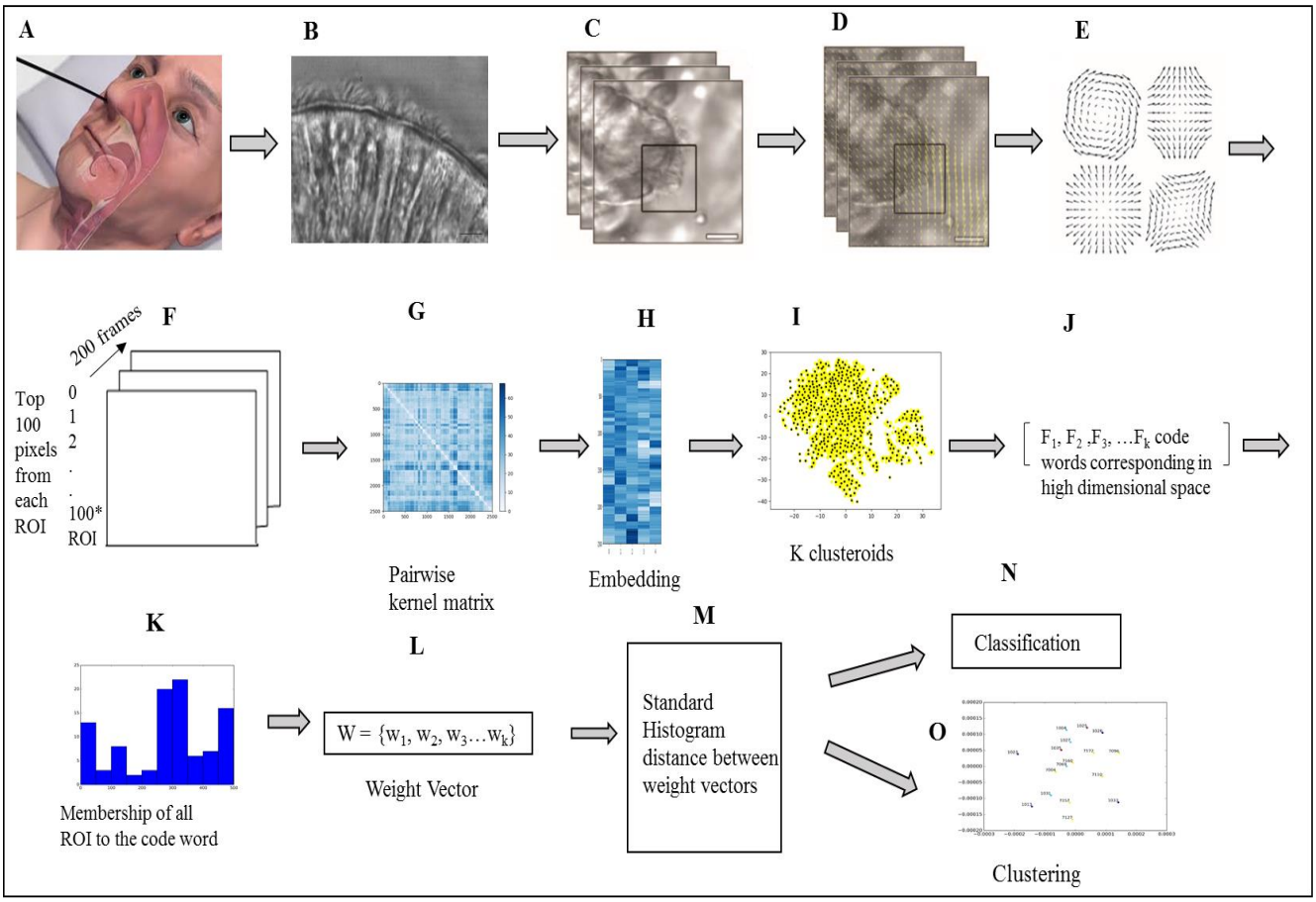

Figure 4: Architecture of our pipeline. (A) Nasal brush biopsy of the patient developed in a petri

dish. (B) CM videos observed in a high speed digital microscope from the patient. (C) ROI drawn on each video, (D) Optical flow computed to observe the direction and magnitude of CM. (E) 3 elemental components of optical flow out of which we used rotation data for further analysis. (F) Picked top 100 pixels with high standard deviation from each ROI across 200 frames video. (G) Computed pairwise kernel matrix between all the ROIs for all patients. (H) Pairwise kernel matrix in non-Euclidean distance is converted to low dimensional embedding by applying nonlinear dimensionality reduction techniques. (I) Applied k means clustering algorithm on the embedded matrix to find k cluster centers. (J) Computed k code words from cluster centers corresponding to original high dimensional space. This is the codebook. (K) Membership of original ROIs with respect to each code word is computed. (L) Represented membership as weight vectors using TF and TFIDF format. (M) Computed the distances between weight vectors using χ^2 *and Cosine similarity. (N) Applied classification algorithms on the kernel matrix to test the accuracy of prediction. (O)Applied clustering algorithms to find the groups of patients closer to each other.*

4.3 Software

In our work, we used Python 2.7 to implement the analysis pipeline. We used joblib for parallelization, and several of the scientific computing packages (NumPy, SciPy, scikit-learn, matplotlib) for reading, analyzing, and serializing the data. We also used the plotting package Matplotlib.

Since we used large data structures, computing time and code efficiency played a major role in developing the computational pipeline. Hence, we used an on-premise BlueData cluster with

large quantities of compute resources to run our code and gather our results. We used cProfile profiling tool in python to perform time analysis and make it more efficient.

CHAPTER 5

RESULTS AND ANALYSIS

This section explains results obtained in our analysis and some interpretations based on the results.

5.1 Identifying Pixels

As explained in chapter 4.1, we used the elemental components of the optical flow (i.e. rotation quantity) instead of grayscale pixel intensities for our analysis. From each ROI, we chose 100 pixels with the largest variation in rotation magnitudes.

Figure 5 illustrates the spatial representation of the pixels we chose using this selection strategy. One frame from each ROI is shown; the x axis is the width of each frame and y axis represents the height of that frame. The pixels chosen likely include some extreme outliers that generate noise; few pixels we selected overlapped with regions containing cilia, and some of them consisted of noise: background and cell bodies. We believe that the background and cell body have high magnitude variations because of translational motion of the sample and the background particulate obstructing the proper view of cilia. Another reason is because of cell body that moves due to the stroke that is caused by CM which is misinterpreted as cilia.

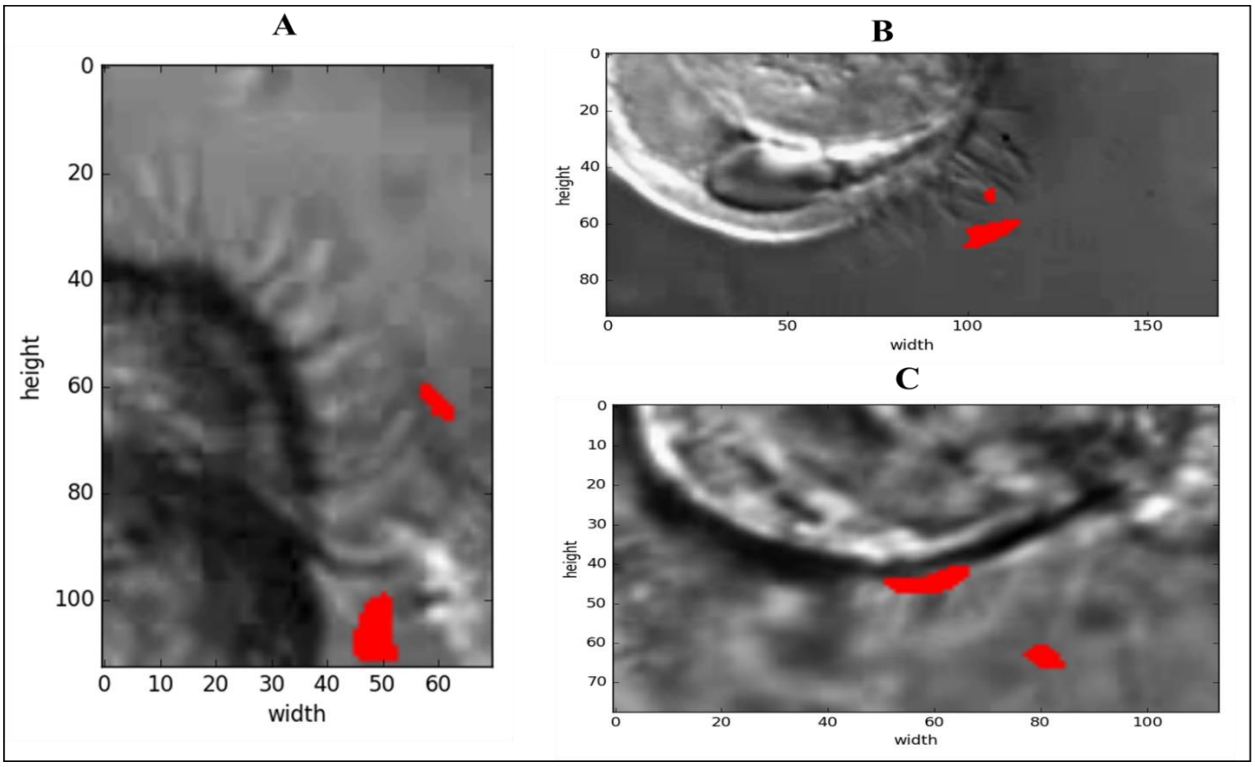

Figure 5: Spatial representation of pixels chosen for a patient. A) The red small patch illustrates the pixels chosen at the edge of cilia and big red patch indicates the cilia. B) For this patient, few pixels chosen are cilia and the big patch is back ground of ROI. C) Big patch indicates the pixels proximal to the cell wall of cilia and other pixels are chosen from the background of ROI.

The temporal representation of these pixels chosen is represented in the form of a kymograph. A kymograph explains whether the motion of pixels across the frames is continues over time. Figure 6 shows the kymograph for 2 patients. It illustrates the time series of the pixels along x axis, where the frames are stacked on top of each other (on y axis). We can see a continuous flow of motion from one pixel to another, while pixels drawn from separate regions are clearly stratified.

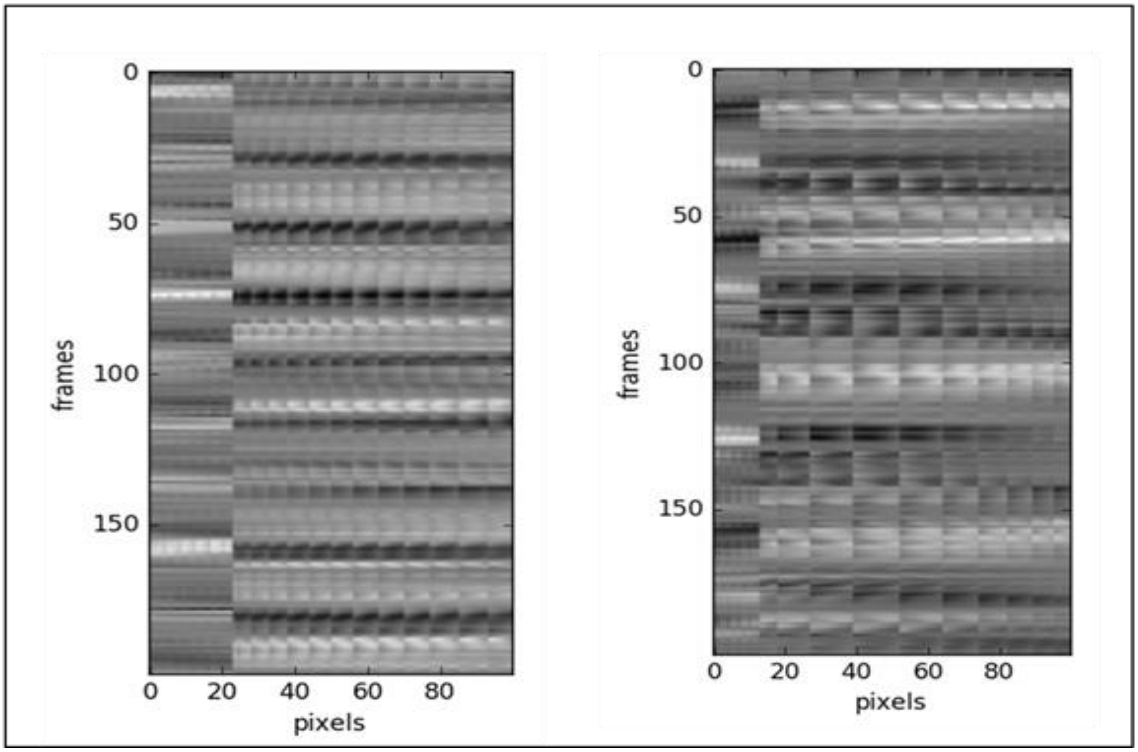

Figure 6: Kymograph representation of the pixels chosen from 2 patients.

5.2 Kernel representation of the patients

We plotted the pairwise distance between each pixel for a patient. After choosing the high magnitude variation pixels, we represented the kernel matrix for each patient using all three-distance metrics. Figures 7 - 9 represents the intra-patient pixel distance for one of the patients using these three-distance metrics. The x axis and the y axis represent the 100 pixels chosen for an ROI for a patient with the ID '1026'. We can see three distinct blobs, labeled A, B, and C in the kernel matrix. These labeled blobs correspond to the chosen regions of pixels as represented in Figure 7. Note how these "blobs" of pixels create a block wise structure in the kernel matrix, where each pixel within a blob is relatively "closer" to the other pixels in the same blob, and relatively "farther" away from the pixels in the other blobs. These 3 blobs can be thought of as 3

cluster patterns that this patient exhibits. We observed similar pattern for this patient using all 3-distance metrics revealing some structure to CM that this patient exhibits. These blobs also correlate with the pixels chosen shown in Figure 10. The red mark in Figure 10 indicates the pixels chosen that might correspond to A, B, and C blobs in figure 7, 8, and 9.

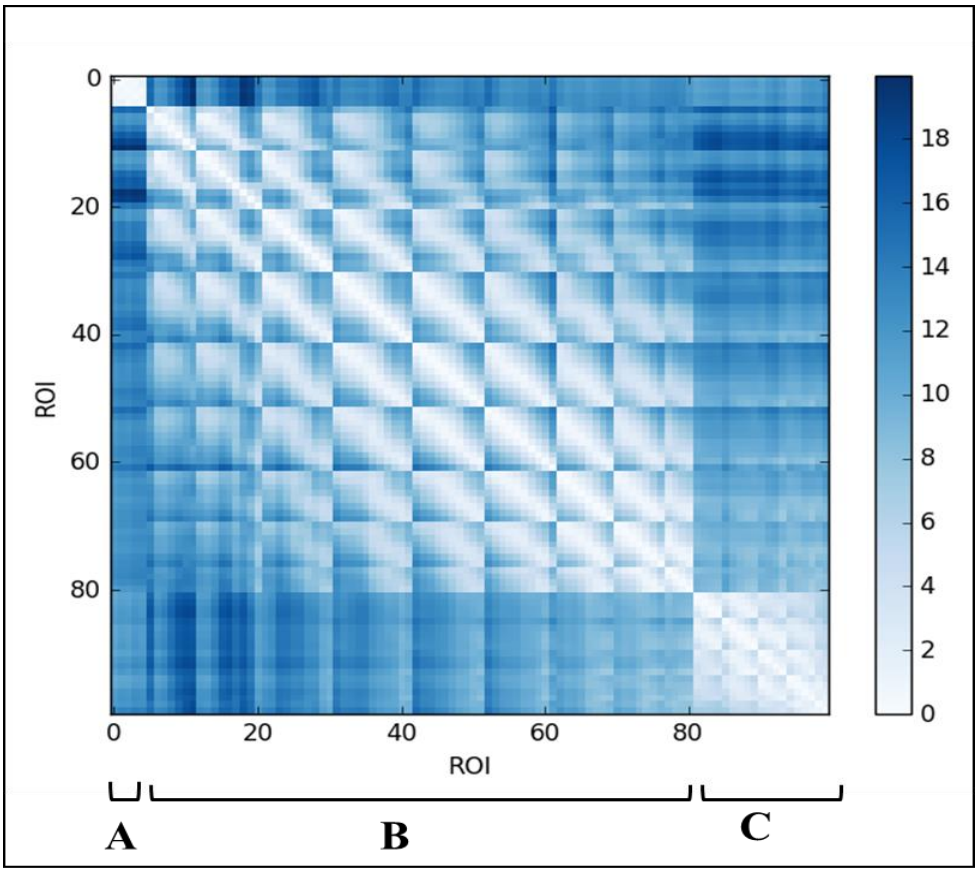

Figure 7: Represents the intra patient pair wise pixel distance computed using Bhattacharya distance for patient '1026'. Right side bar represents the color coding based on distance between each pair. 0 distance represents the distance with the pair itself and is white in color. Darker color represents that the distance between the pair is higher.

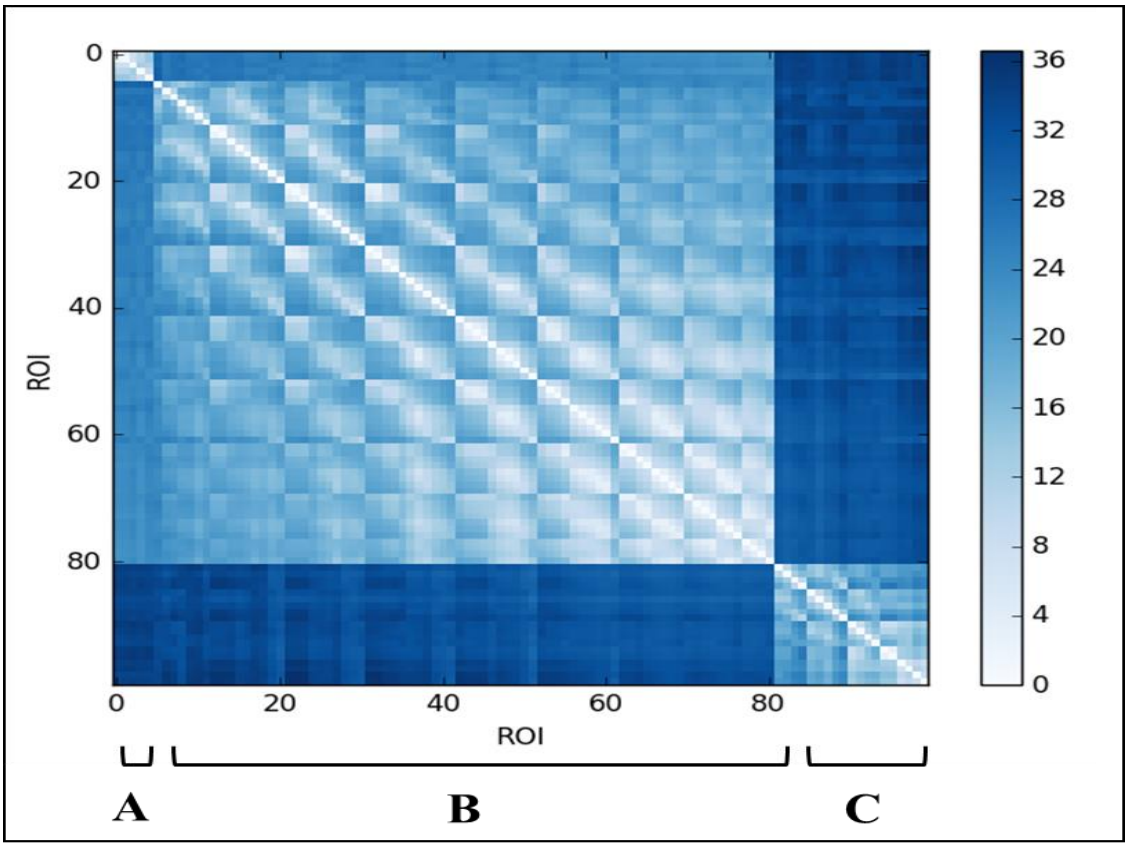

Figure 8: Represents the intra patient pair wise pixel distance computed using cepstral distance for patient '1026'. A, B, C reveals the CM structure this patient exhibits.

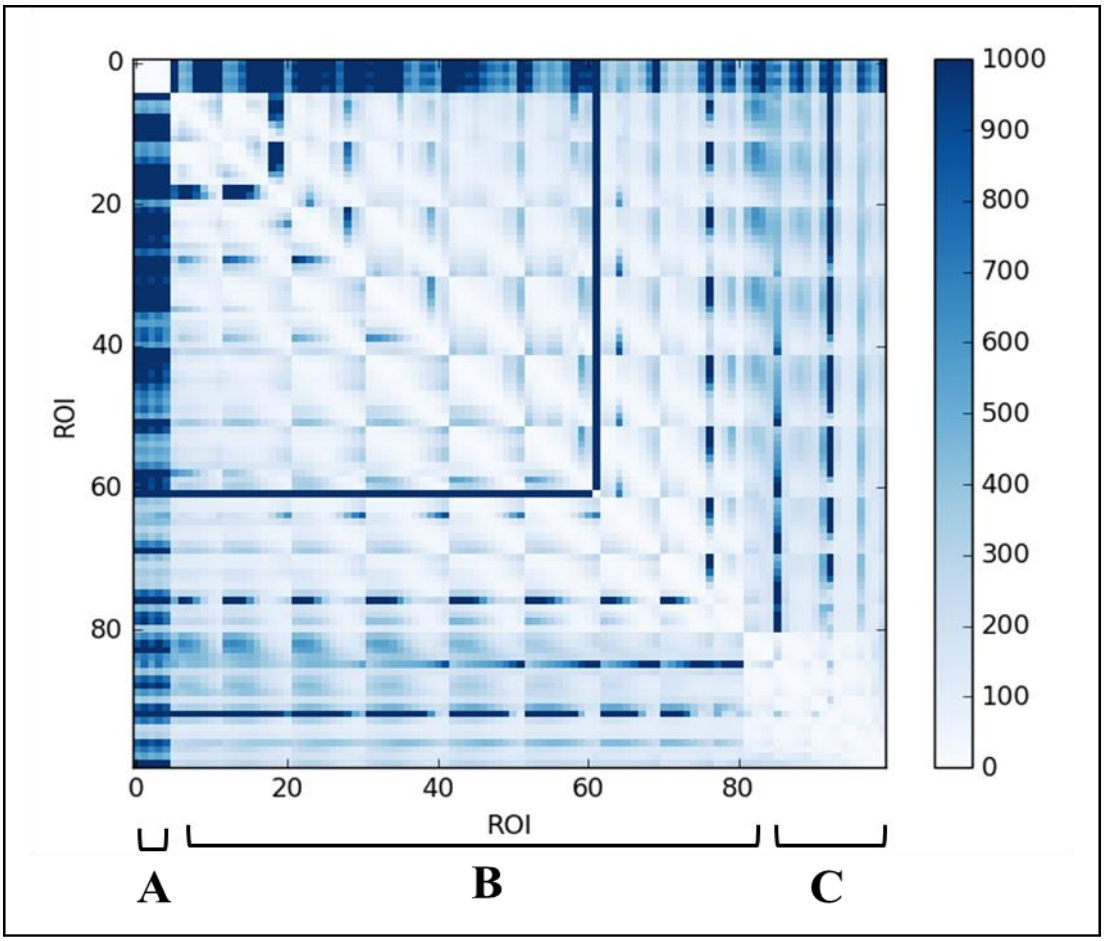

Figure 9: Represents the intra patient pair wise pixel distance computed using KL divergence for patient '1026'.

Figure 10 shows the spatial representation of the pixels chosen for patient '1026'. It shows that most of pixels chosen are cilia and few pixels are chosen from the background.

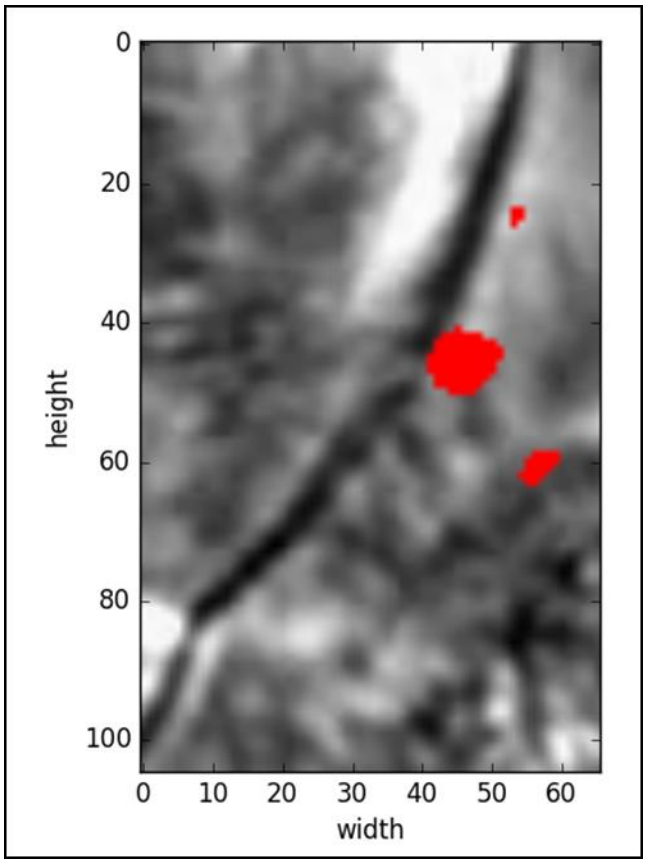

Figure 10: Spatial representation of pixels chosen for patient '1026'.

Consider another patient with ID '7127'. This patient has four ROIs, and the spatial representation for the pixels chosen is shown in Figure 11. It indicates that most of the pixels chosen are from the background of the video and only a few of them are from cilia. The pairwise distance between the pixels chosen for the four ROIs is computed using Bhattacharya distance, Cepstral distance and KL divergence (Figures 12-14). The kernel matrix shows a pattern of seven distinct groups using all three distances. Although Figure 11 shows that background pixels are chosen, there is still a common pattern that these distance metrics reveal.

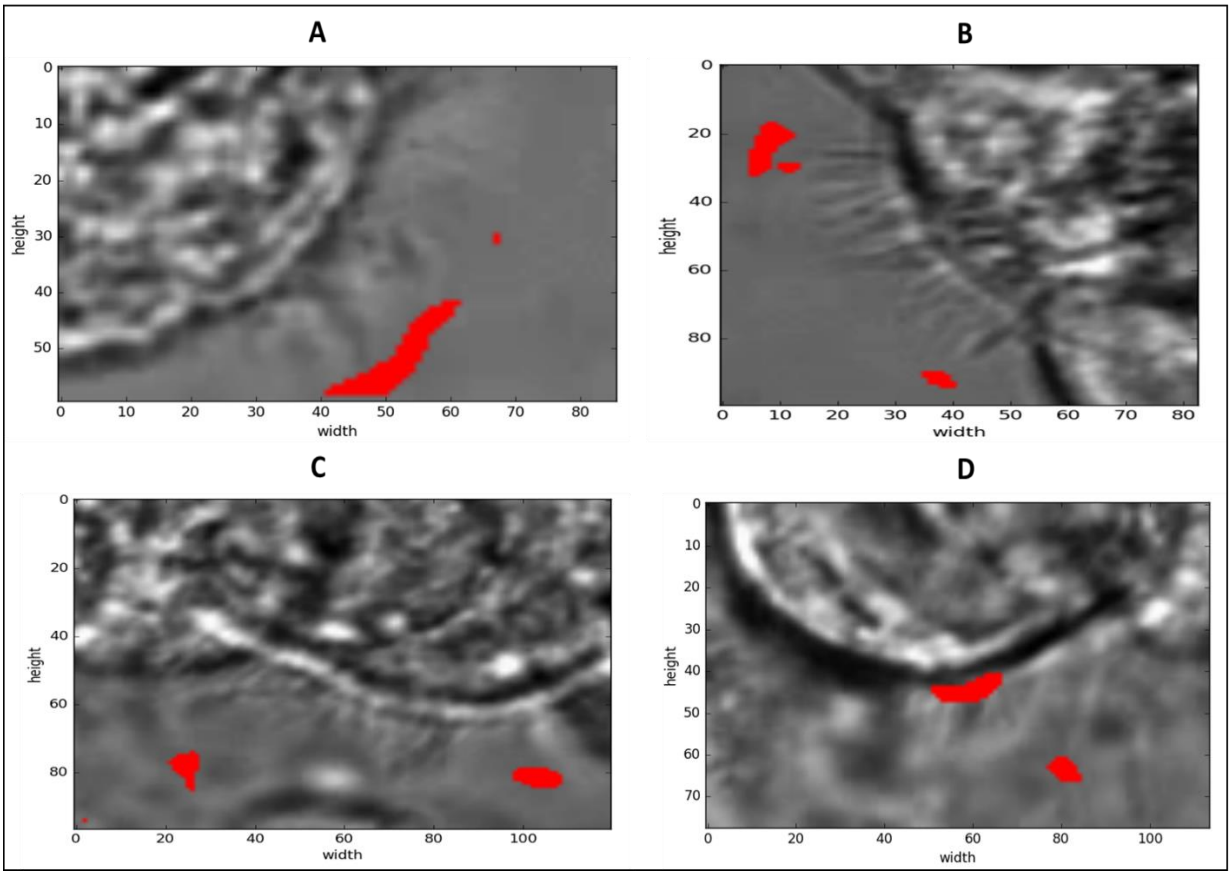

Figure 11: Spatial representation of the pixels picked for patient '7127'. A, B, C, D are 4 ROIs chosen for this patient and red color represents the pixels chosen.

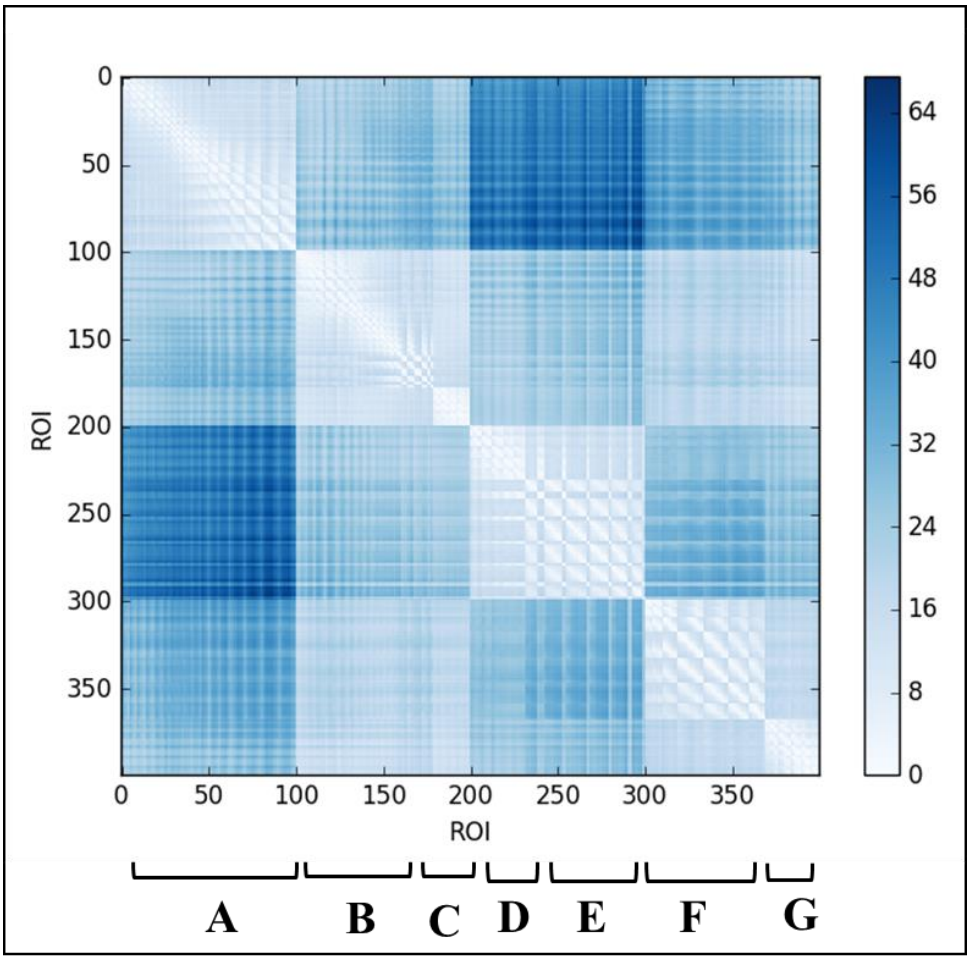

Figure 12: Pairwise distances computed between pixels using Bhattacharya distance for patient '7127'

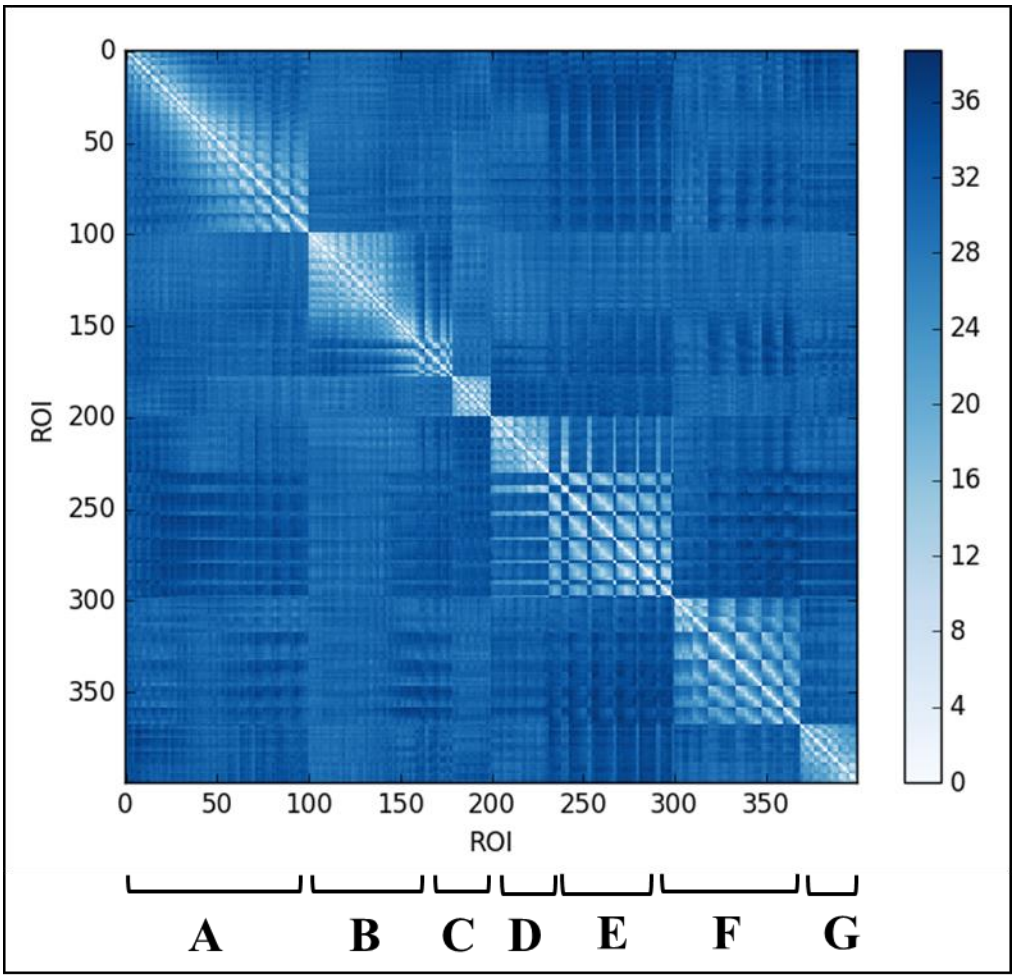

Figure 13: Pairwise distances computed between pixels using cepstral distance for patient '7127'

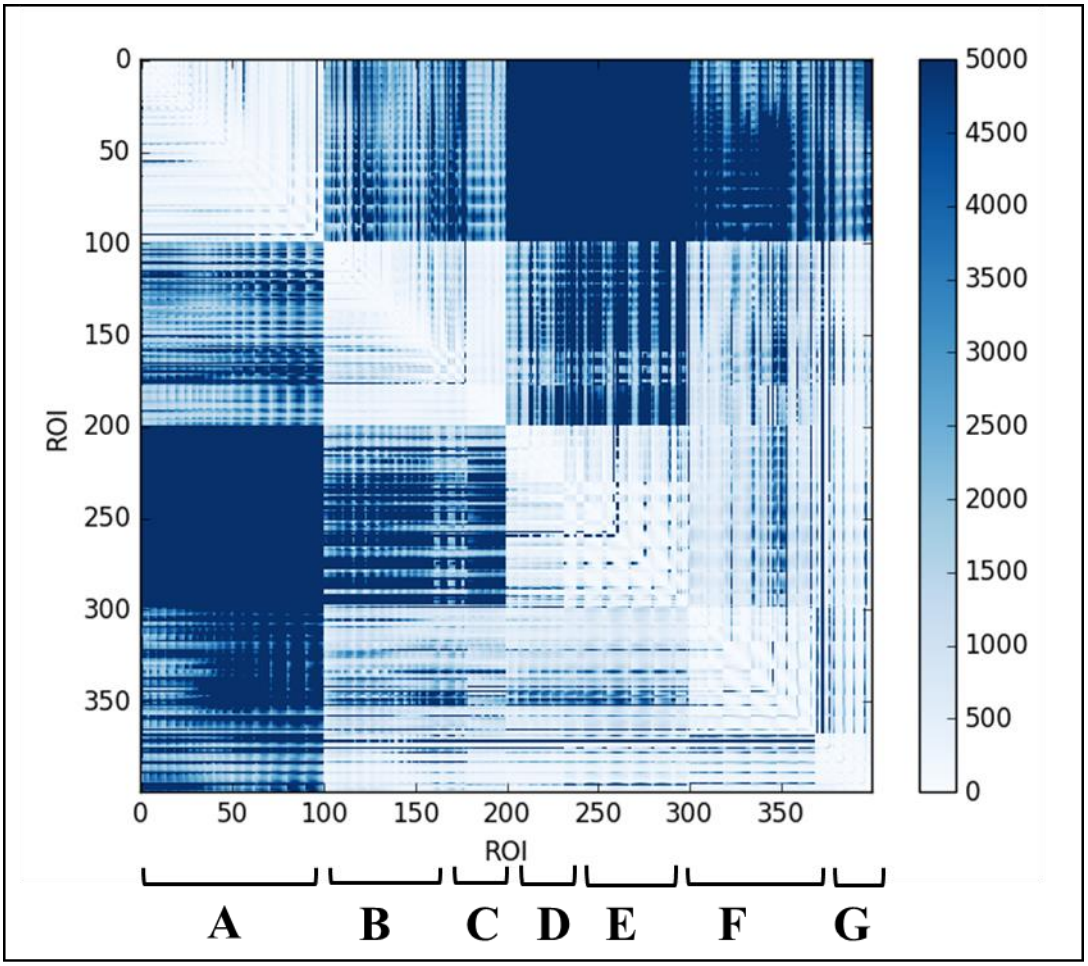

Figure 14: Pairwise distances computed between pixels using KL divergence for patient '7127'

5.3 Clustering Results

Table 2 represents the different combinations of the results obtained using three distance metrics, dimensionality reduction techniques, and patient weight representations. We represented patient weight vectors using both TF and TFIDF. But here we provide the cluster representation for only TF weight vectors for all the distance metrics.

Distance Metric	Embedding	Patient weight representation method	Figure number
Bhattacharya distance	Isomap	TF	Figure 15
	LEM	TF	Figure 16
	MDS	TF	Figure 17
	PCA	TF	Figure 18
KL divergence	Isomap	TF	Figure 19
	LEM	TF	Figure 20
	MDS	TF	Figure 21
	PCA	TF	Figure 22
Cepstral distance	Isomap	TF	Figure 23
	LEM	TF	Figure 24
	MDS	TF	Figure 25
	PCA	TF	Figure 26

Table 2: Represents the cluster figure numbers formed using different combinations.

Figures from 15 – 18 are the plots of the clusters for Bhattacharya distance metric. The numbers in the cluster represents patient names. The color-coded dots represent the 4 clusters formed after applying spectral clustering. Patients belonging to same cluster are coded with same color. Note that these figures are plotted using the t-SNE after reducing the dimensions of the kernel matrix to 2 and plotted using scatter plot.

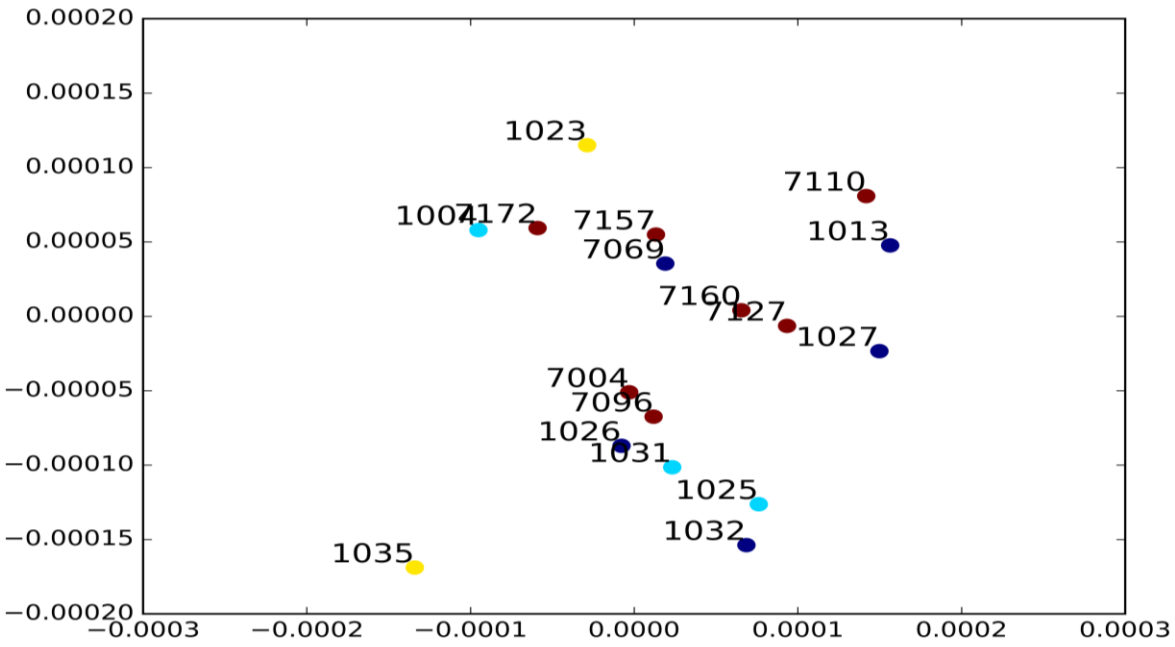

Figure 15: Cluster results formed using Bhattacharya distance and isomap embedding represented using TF weight vector.

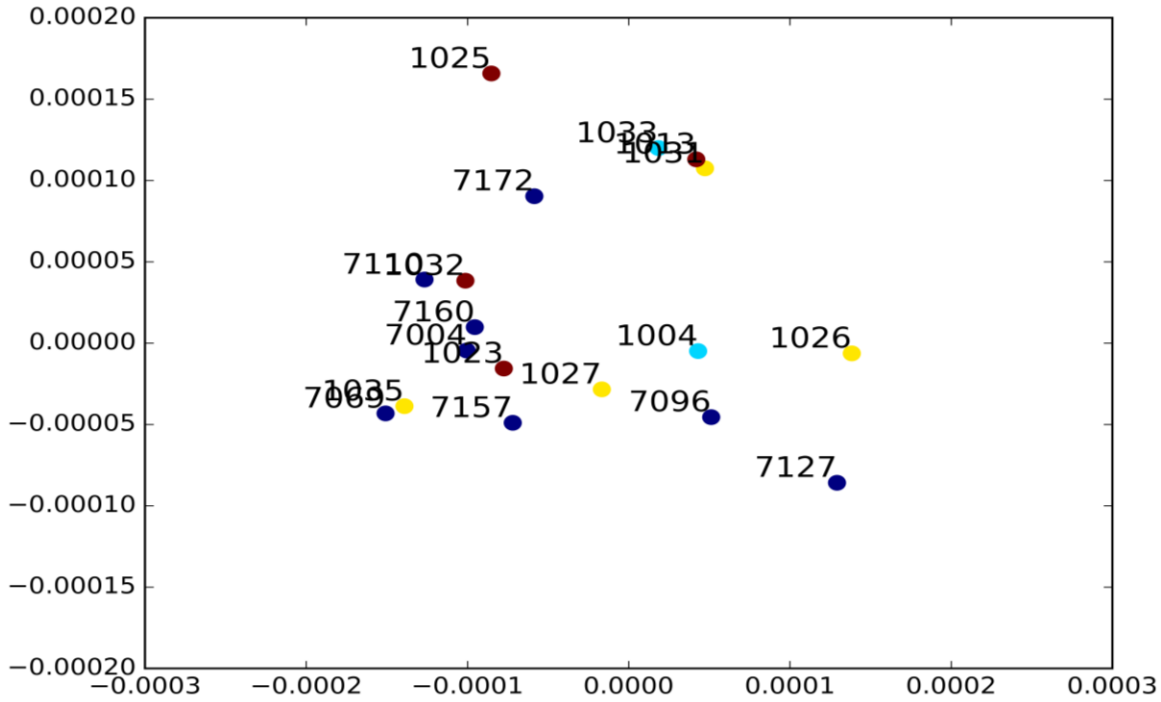

Figure 16: Cluster results formed using Bhattacharya distance and LEM embedding represented using TF weight vector.

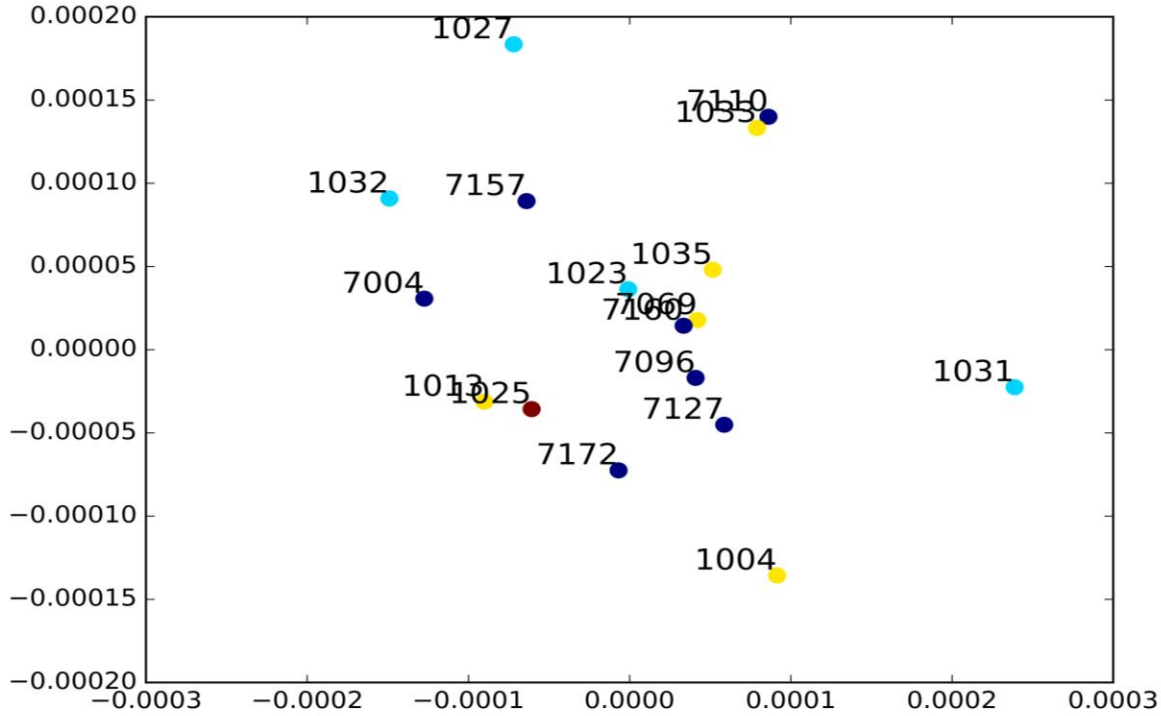

Figure 17: Cluster results formed using Bhattacharya distance for pairwise kernel and MDS embedding represented using TF weight vector.

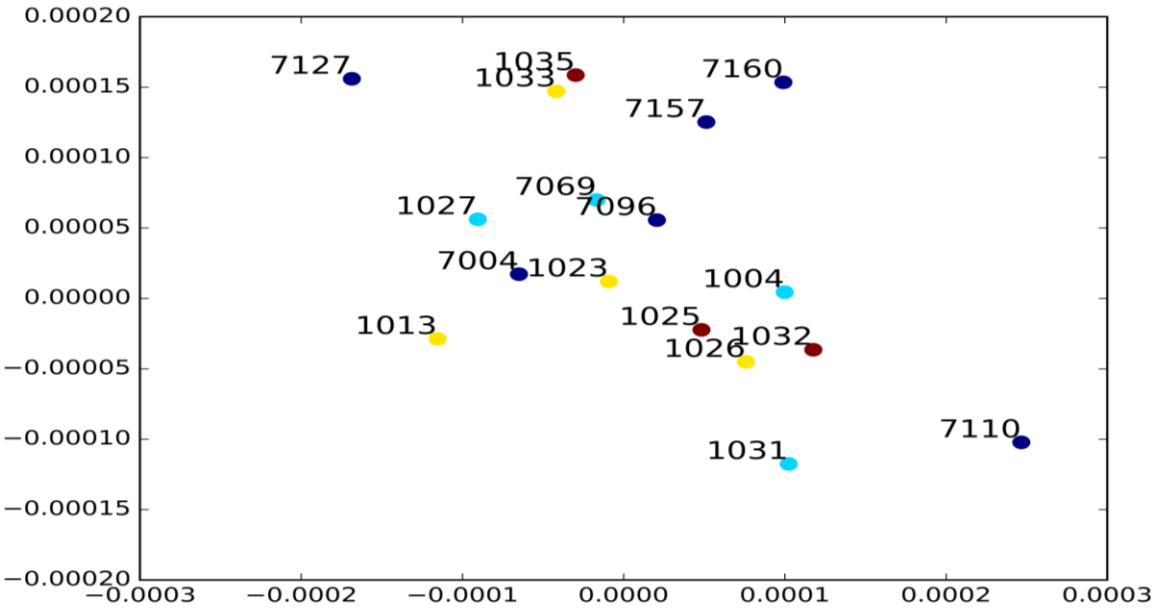

Figure 18: Cluster results formed using Bhattacharya distance and PCA embedding represented using TF weight vector.

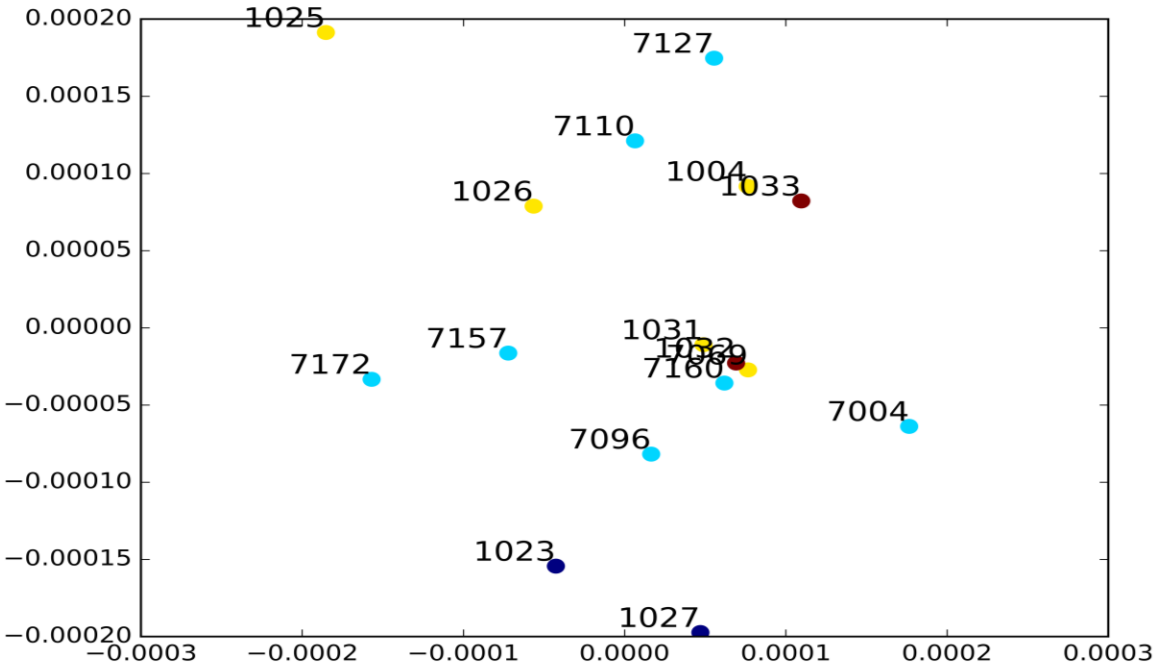

Figure 19: Cluster results formed using KL divergence and Isomap embedding represented using TF weight vector.

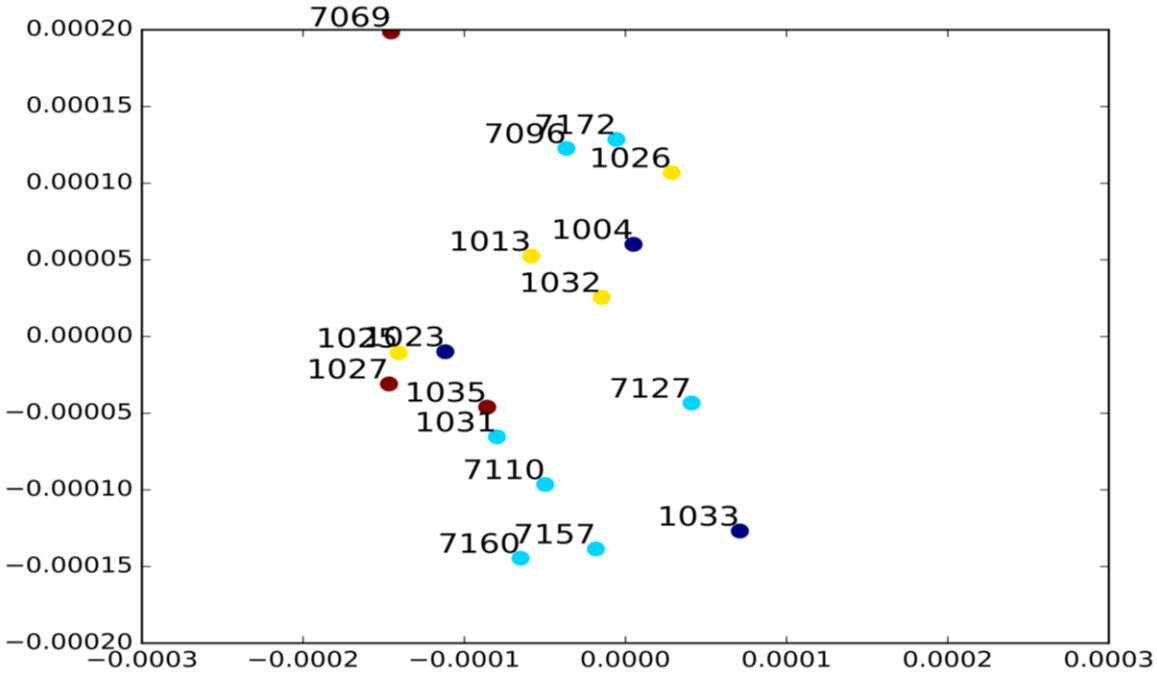

Figure 20: Cluster results formed using KL divergence and LEM embedding represented using TF weight vector.

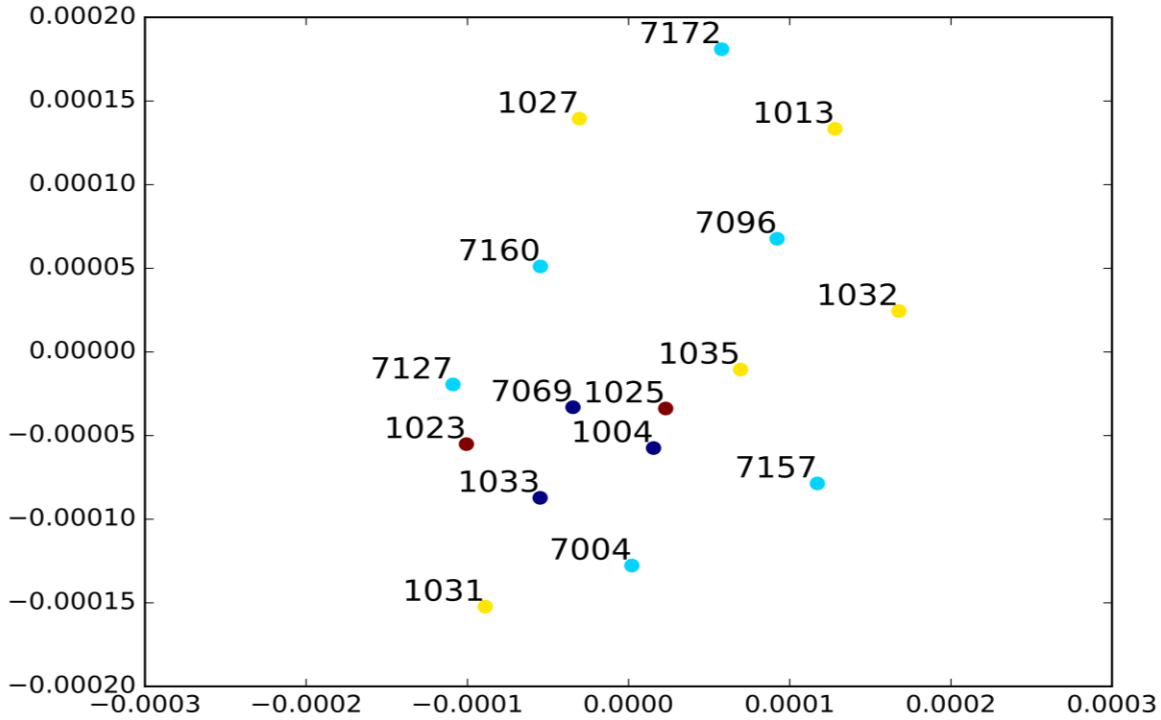

Figure 21: Cluster results formed using KL divergence and MDS embedding represented using TF weight vector.

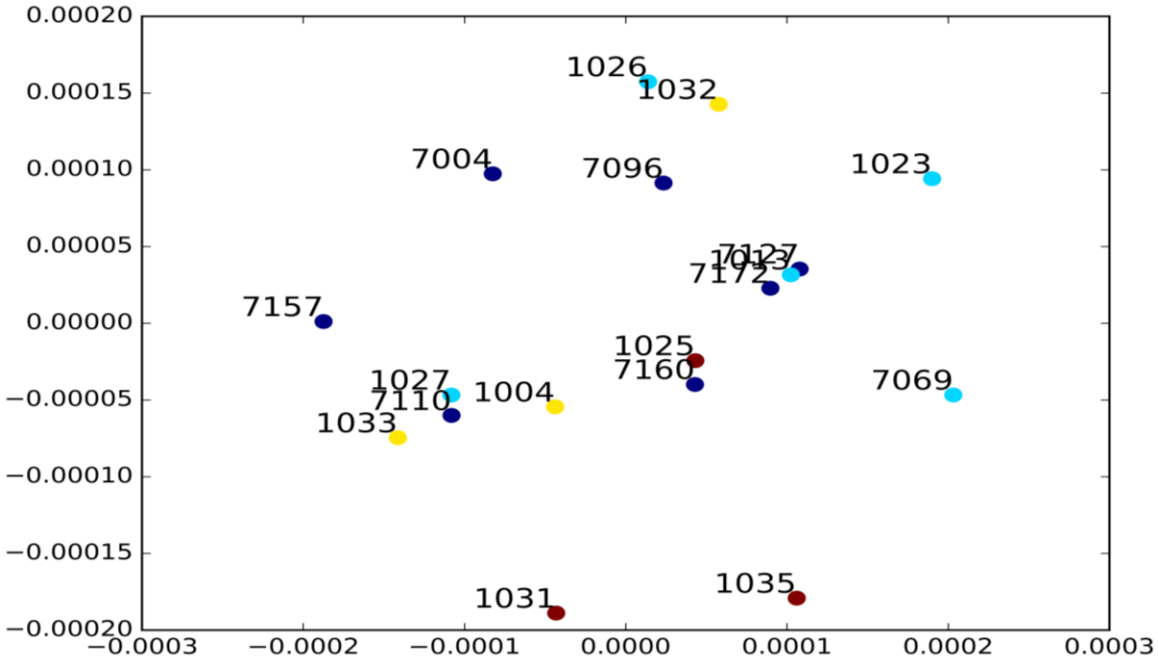

Figure 22: Cluster results formed using KL divergence and PCA embedding represented using TF weight vector.

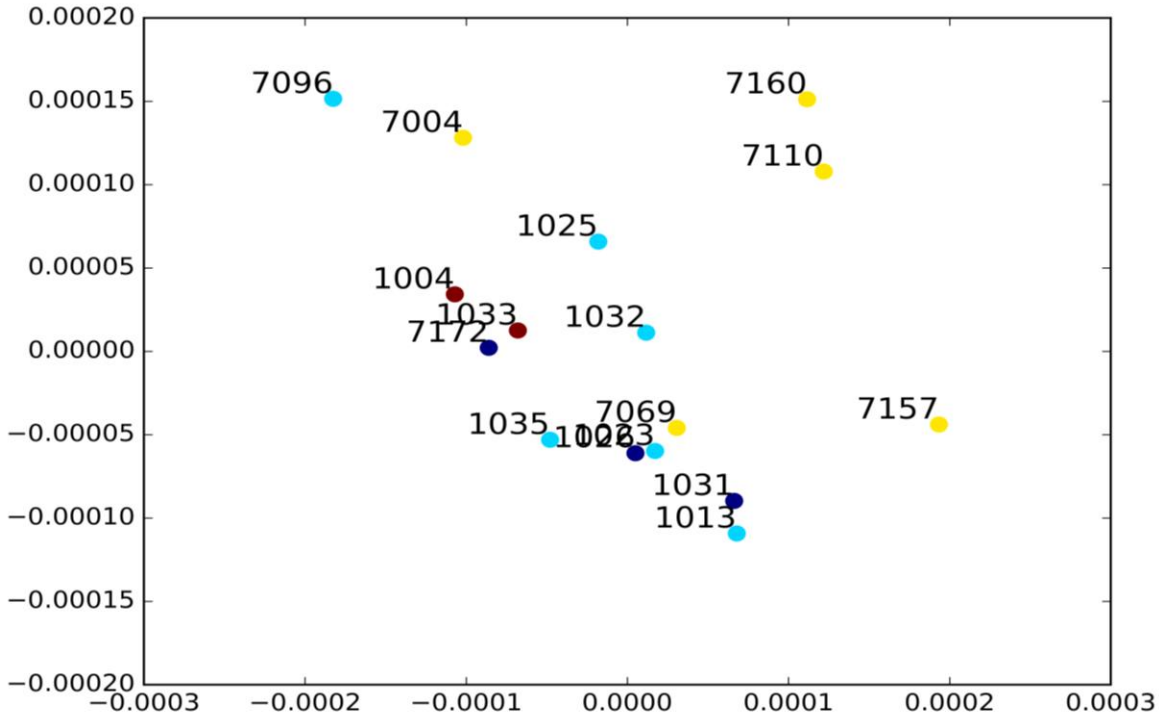

Figure 23: Cluster results formed using Cepstral distance and Isomap embedding represented using TF weight vector.

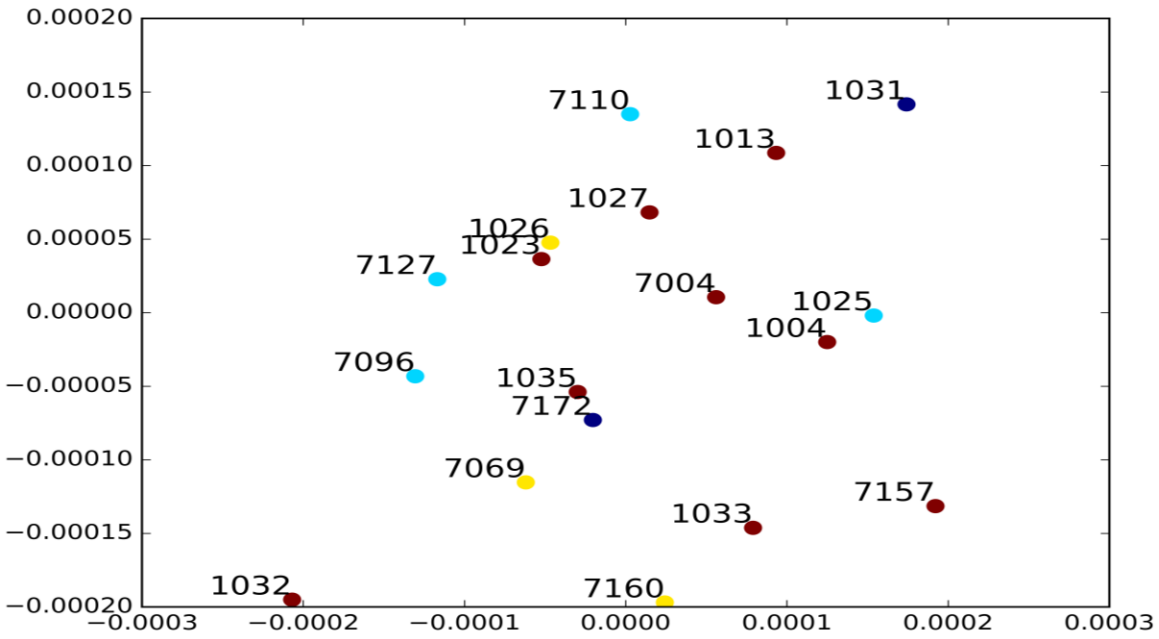

Figure 24: Cluster results formed using Cepstral distance and LEM embedding represented using TF weight vector.

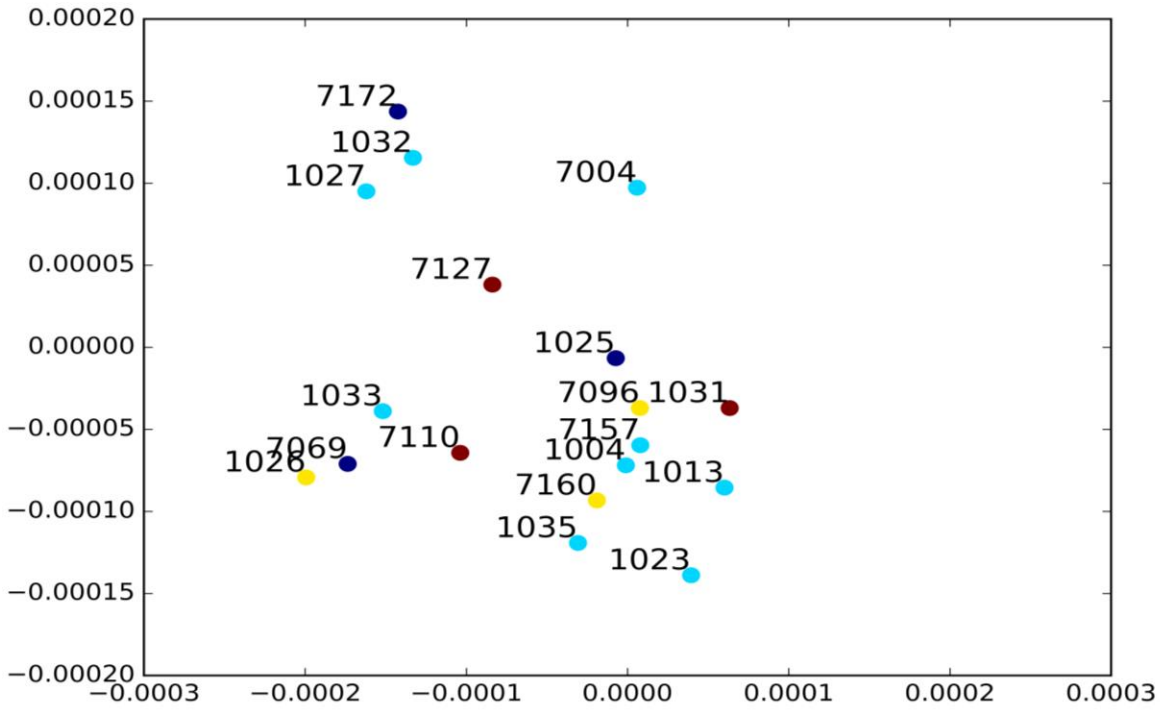

Figure 25: Cluster results formed using Cepstral distance and MDS embedding represented using TF weight vector.

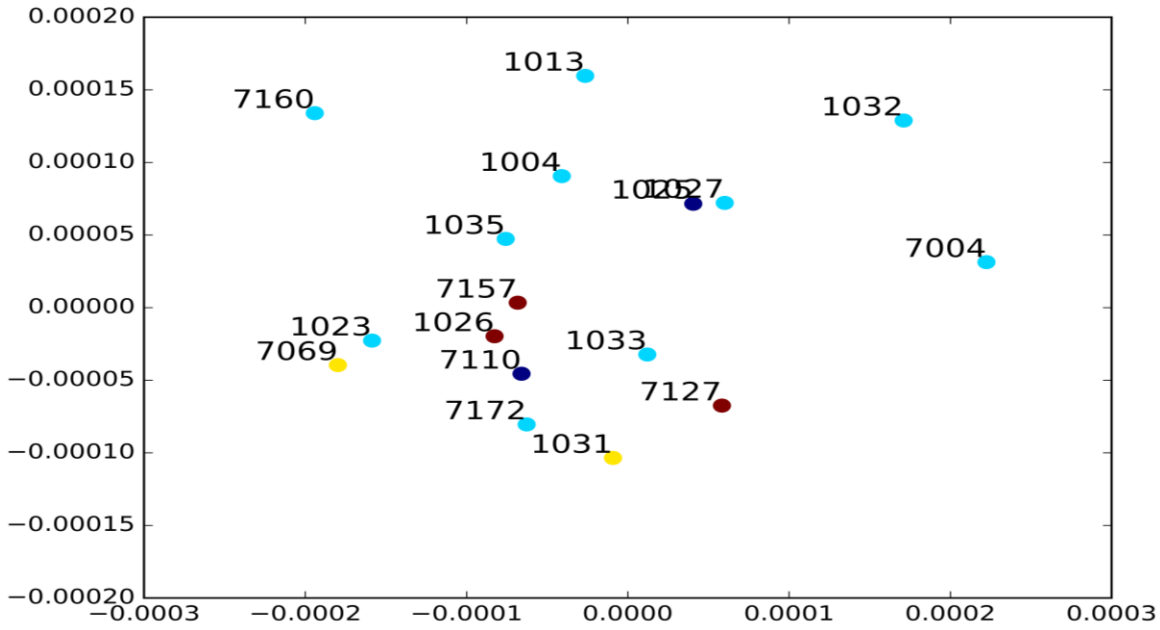

Figure 26: Cluster results formed using Cepstral distance and PCA embedding represented using TF weight vector.

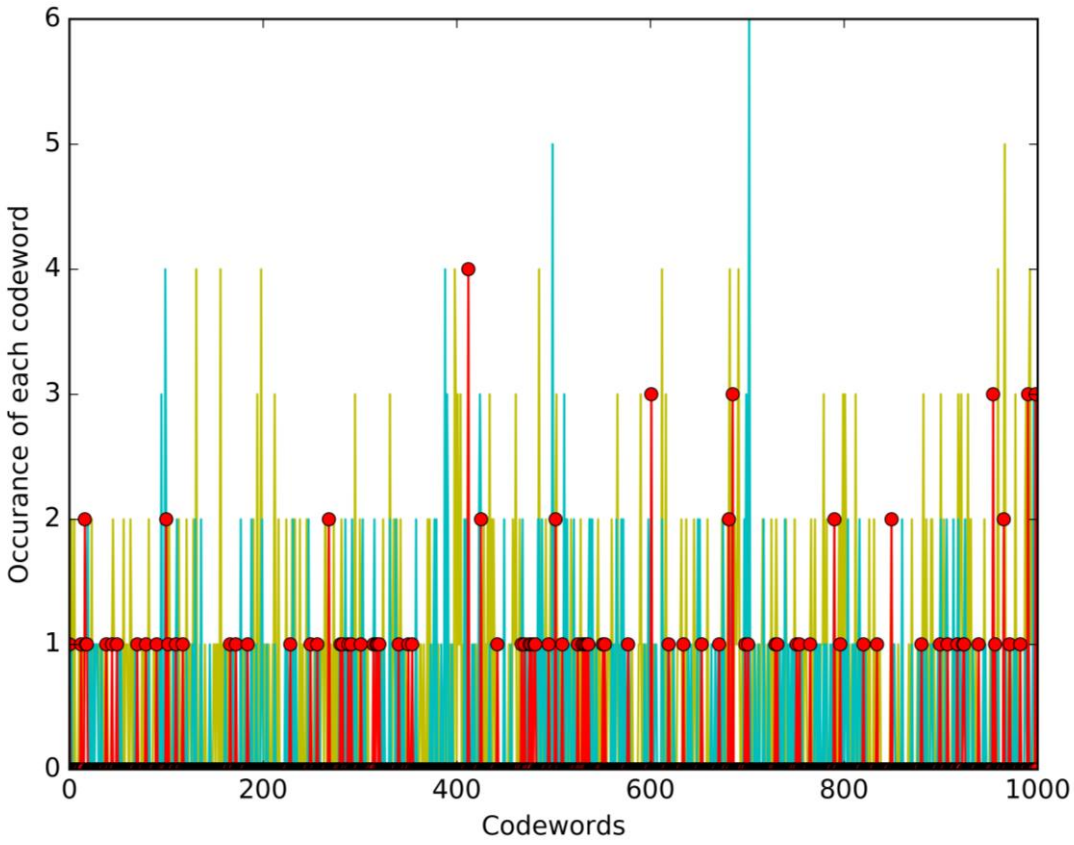

Figure 27: Distribution of codewords between patients 7096 and 7127. X axis indicates 1000 code words and y axis represents the number of occurrences of each codeword for a patient. Yellow line indicates the codeword distribution for patient 7096 and cyan represents the codeword distribution for patient 7127. Red indicates the codewords common for both the patients.

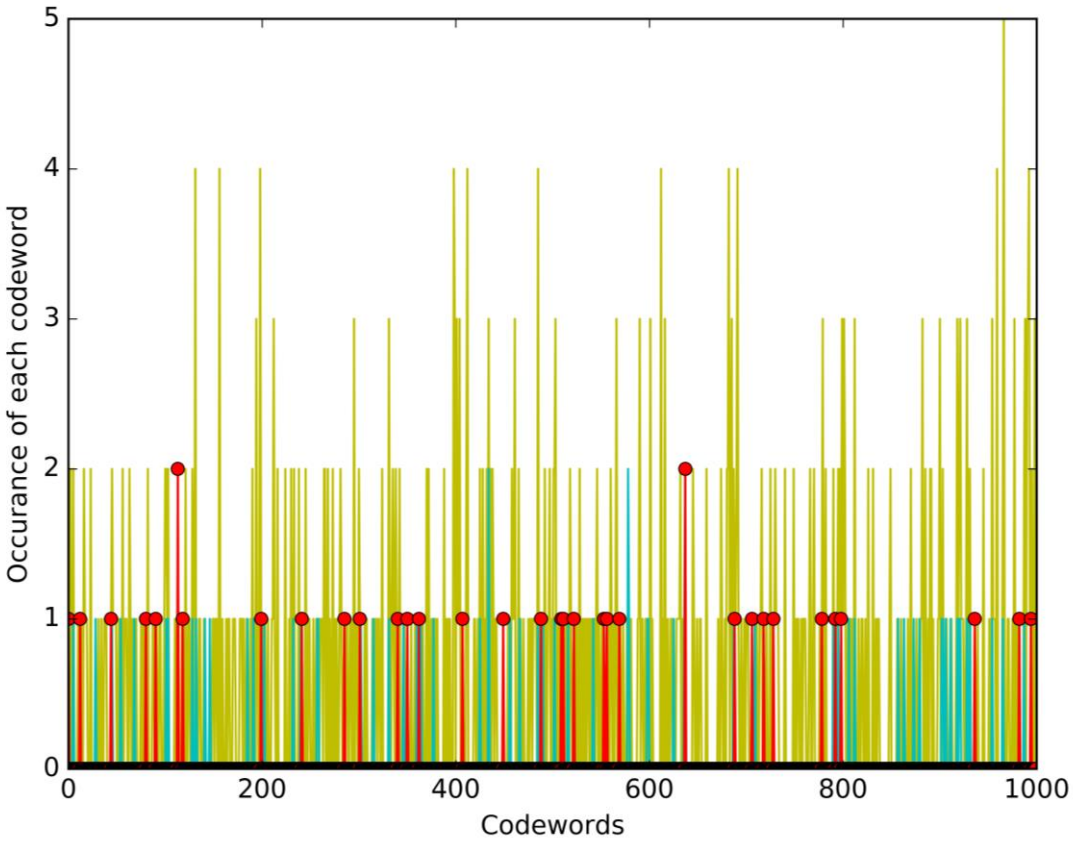

Figure 28: Distribution of codewords between patients7096 and 1035. X axis indicates 1000 code words and y axis represents the number of occurrences of each codeword for a patient. Yellow line indicates the codeword distribution for patient 7096 and cyan represents the codeword distribution for patient 1035. Red indicates the codewords common for both the patients.

Clustering results analysis

From the above clustering results, we make the following observations.

5.3.1 Observations using Bhattacharya and KL divergence kernel

Clustering results obtained for both Bhattacharya and KL divergence are similar for many patients. We observed that some patients are always clustered together and some clustered differently in both distance metrics.

The formula for computing KL divergence using equation 6 can be re written as

$$D_{KL}(F1, F2) = \sum_{0<vk<\frac{1}{2}} [trace\left\{F_1(V_k)\frac{1}{|F_2(V_k)|}adjuct(F_2\,(V_k)\right\} - \ln\frac{|F_1(V_k)|}{|F_2(V_k)|} - N] \qquad (25)$$

Comparing this to computing Bhattacharya distance in equation 7 and replacing α with 0.5 gives

$$D_B(\alpha, F1, F2) = \frac{1}{2}\sum_{0<vk<\frac{1}{2}} [\ln\frac{1}{2}\frac{|F_1(V_k)+F_2(V_k)|}{|F_2(V_k)|} - \frac{1}{2}\ln\frac{|F_1(V_k)|}{|F_2(V_k)|}] \qquad (26)$$

Comparing equations 25 and 26, they both differ in

$$D_{KL}(F1, F2) = \sum_{0<vk<\frac{1}{2}} [trace\left\{F_1(V_k)\frac{1}{|F_2(V_k)|}adjuct(F_2\,(V_k)\right\} - \text{constant}]$$

$$D_B(\alpha, F1, F2) = \frac{1}{2}\sum_{0<vk<\frac{1}{2}} [\ln\frac{1}{2}\frac{|F_1(V_k)+F_2(V_k)|}{|F_2(V_k)|} - \ constant]$$

Above equations differ only in first part of computation. *F(v)* for both the distance metrics is computed using time series method in equation 5. Hence, this might be the reason that patient cluster results produced by both these distance metrics are similar.

1. Patients 7110, 7157, 7160, 7172, and 7004 are mostly clustered together for both the distance metrics. The spatial representation of the pixels chosen for all these patients contains both cilia and background of cilia. The clinicians labeled patients 7110 and 7172 as two and patients 7157, 7160, and 7004 are marked as four. Since noise is chosen in addition to cilia, we cannot predict with confidence that these patients exhibit similar CM patterns.
2. Patients 7127 and 7096 are grouped together in most cases for both distance metrics. Most of the pixels chosen for patient 7096 are cilia and very few pixels are from the

background. For patient 7127, pixels are chosen from both cilia and background. Both these patients are marked three by the clinicians. We can speculate that the CM patterns for both these patients are probably similar since most of the pixels analyzed are of cilia. We chose six ROIs for patient 7096 and four ROIs for patient 7127. There are at least seventy code words that are expressed at a similar level for both these patients. Using PCA embedding, nine code words commonly appeared twice in both the patients and five code words commonly appeared thrice. In case of LEM, thirteen common code words appeared twice in both the patients and two code words commonly appeared thrice. Figure 27 shows the code word distribution common for both the patients.

3. Patients 7127 and 1035 are always clustered in different groups. The data analyzed for patient 1035 contains cilia whereas the pixels chosen for patient 7127 consists of both cilia and background. Patient 1035 is marked as one by the clinicians and patient 7217 is marked as three. We have four ROIs for patient 7127 and one ROI for patient 1035. Code word distribution of these patients show that there are maximum twenty-seven code words that are expressed at the same level and all these code words are expressed just once for both the patients. From the above observations, we can probably believe that these patients might have different CM patterns. But, we cannot arrive at a conclusion since the data chosen for patient 7127 contains background along with cilia.
4. Patients 7096 and 1035 are grouped in different clusters. The data analyzed for patient 1035 contains only cilia and for patient 7096 contains mostly cilia and very few pixels from the background. Patient 7096 is marked as three and patient 1035 is marked as one by the clinicians. We have six ROIs for patient 7096 and one ROI for patient 1035. The code words commonly expressed for these patients using MDS embedding are twenty-

three and for PCA are thirty-four. Twenty-one code words out of twenty- three and thirty-two out of thirty-four code words occurred once. Two code words occurred twice. Based on the above analysis, we can speculate that these patients might exhibit different CM patterns, hence clustered differently by two distance metrics. Figure 28 represents the common code words expressed between patients 7096 and 1035. Yellow lines, which represents the codeword distribution for patient 7096 are higher compared to the cyan lines because the number of ROIs for patient 7096 are more compared to patient 1035.

5. When Bhattacharya distance is used for computing the pairwise kernel, patients 1013 and 7069 are clustered together in most cases. The pixels chosen for patient 1013 contains mostly background and cell body of the cilia, whereas for patient 7069 mostly background and small portion of cilia is chosen for our analysis. Patient 1013 is marked as one by the clinicians and patient 7069 is marked as four. We cannot conclude anything based on the results since the data we analyzed is noisy.
6. Using KL divergence for computing kernel, patients 1025 and 1026 are clustered together. The data chosen for both these patients contains cilia. Both these patients are marked as one by the clinicians. We have one ROI for each patient. There are eleven common code words for both these patients using MDS embedding. Since the pixels chosen do not contain noise, we can speculate that these two patients might exhibit similar CM pattern.
7. Patients 1027 and 1013 are also clustered together using KL divergence. The data analyzed for patient 1027 contains mostly cilia and very few pixels from the background, whereas pixels chosen for patient 1013 contain mostly noise. Both these patients are

labeled as one by the clinicians. Due to noise, we cannot conclude that these patients have similar CM patterns.

5.3.2 Observations using cepstral distance

We computed the cepstrum of the time series using inverse DFT of logarithm of DFT of each component rather than using the time series method. Hence this metric produced different clustering results compared to the above time series distance metrics.

1. Patients 1032, 1013, and 1035 are clustered together. The data chosen for patient 1013 and 1032 contains mostly noise and only few cilia, whereas the pixels chosen for patient 1035 contains only cilia. All these patients are marked as one (normal) by the clinicians. Since the data, we analyzed is noisy, we cannot conclude that these two patients exhibit similar CM pattern.
2. Patients 7110 and 7127 are clustered together in most cases. Most of the pixels chosen for both these patients are background while few of them are cilia. Patient 7110 is marked as two and patient 7127 is marked as three by the clinicians. Since the data, we chose mostly consists of noise, we cannot conclude if these patients have similar CM pattern.
3. Patients 7127 and 1035 are always clustered in different groups. The pixels chosen for patient 7127 are mostly background and few pixels are cilia, whereas the pixels picked for patient 1035 are cilia. Patient 7127 is labeled as three and patient 1035 is labeled as one by the clinicians. Since the data, we analyzed contains noise along with cilia, we cannot conclude if these patients have different CM.

5.4 Classification results

After computing bag of dynamical systems approach on the cilia, we also applied classification techniques. We used stratified 10- fold cross validation technique and reported the averaged accuracy across these cross validation runs.

The ground truth labels marked by the clinicians are just a measure of the degree of abnormality. Hence, the accuracy of prediction does not yield information about CM subtypes, but helps to deepen our intuition of the data and evaluate our quantitative CM representations. In other words, a patient marked three can exhibit different CM patterns that can be stiff, wavy or controlled. Hence, we cannot decide the CM subtypes based on these labels.

In SVM classifier, we used a precomputed kernel matrix obtained from χ^2 distance and cosine similarity. It produced results with accuracy between 63% to 70 % using all three-distance metrics. The standard deviation for SVM classifier is relatively lower compared to random forest and k-NN. SVM tries to find the best possible hyperplane for the patterns, which are not linearly separable by transformations of original data, to map into new space. It looks at the global picture to identify the hyperplane. Hence overall, we could classify with decent accuracy using SVM.

Bagging classifier too has less standard deviation, compared to k-NN, random forest and ada boost classifiers. The samples are drawn at random with replacement, so the accuracy across different embedding techniques varied due to randomization. The standard deviation of ada boost classifier is relatively higher compared to other classification techniques. We used decision tree as the base estimator. It tries to adjust weights based on each iteration, due to this adaptive model ada boost performed better compared to bagging in some cases.

We can see a lot of variation in accuracy results in case of k-NN. In some cases, it was only able to predict with an accuracy of 18 % where as in other cases, it was able to predict with 68% accuracy. This is because the standard deviation of k -NN is very high. In other words, the data points are distributed unevenly around the mean and because of outliers accuracy fluctuated a lot. K-NN depends on multiple factors like the type of embedding used, initialization of data points and value of k.

Random forest operates on multiple subsamples of the data and averages the results. Hence, the variance is taken care of to some extent if not completely eliminated. We see that in few cases it performed very well with accuracy of 81% but in some cases, it could only predict with 55% accuracy. This is due to the presence of variation in the data points.

Tables 3-5 shows the classification results using multiple embedding techniques across three distance metrics.

	SVM	Random Forest	KNN	Bagging	Ada boost
MDS TF	66.6	61.9	63.3	60	67.5
MDS TFIDF	63.3	73.3	48.3	67.5	63.3
Isomap TF	63.3	73.5	58.3	71.5	61.2
Isomap TFIDF	60	65	35	66.6	63.8
PCA TF	66.6	66.6	65	70	63.3

PCA TFIDF	60	55	35	61.2	57.6
LEM TF	63.3	68.3	68.3	72.2	66.6
LEM TFIDF	70.8	78.3	58.3	79.1	70.8

Table 3: Classification results obtained by computing kernel matrix using Bhattacharya distance metric

	SVM	Random Forest	KNN	Bagging	Ada boost
MDS TF	63.3	68.3	58.3	65	54.1
MDS TFIDF	70	73.3	21.6	56.6	65
Isomap TF	66.6	75	58.3	58.3	70.8
Isomap TFIDF	66.6	66.6	66.7	66.6	66.6
PCA TF	70.8	65.8	55.8	67.9	72.6
PCA TFIDF	63.3	66.6	18.3	70.8	56.9
LEM TF	66.6	80	60	66	66
LEM TFIDF	70.8	81.6	47.5	65	65

Table 4: Classification results obtained by computing kernel matrix using KL divergence.

	SVM	*Random Forest*	*KNN*	*Bagging*	*Ada boost*
MDS TF	*70.8*	*60.8*	*47.5*	*55.7*	*49.3*
MDS TFIDF	*66.6*	*61.6*	*38.3*	*56.6*	*56.6*
Isomap TF	*66.6*	*56.6*	*60*	*55.7*	*42.5*
Isomap TFIDF	*63.3*	*53.3*	*48.3*	*59.7*	*45.8*
PCA TF	*66.6*	*66.6*	*56.6*	*50*	*50*
PCA TFIDF	*66.6*	*66.6*	*65*	*56.6*	*56.6*
LEM TF	*70.8*	*65.8*	*55.8*	*63.8*	*63.8*
LEM TFIDF	*63.3*	*58.3*	*58.3*	*56.6*	*30*

Table 5: Classification results obtained by computing kernel matrix using cepstral distance.

From the above clustering and classification result analysis, we found that the analyzed data contains both noise and cilia. Hence, we cannot conclude similarity and dissimilarity between patients based on these results. To eliminate this problem, we manually removed the ROIs where noise is chosen. We are then left with 15 patients with 25 ROIs. We removed two patients that are labeled one and one patient that is labeled four. These 25 ROIs contain pixels with only cilia. We re-applied our entire pipeline on this refined ROI. Since the number of ROIs we experimented with after refining the data reduced, we used 500 code words instead of 1000. The

following section analyzes the results produced on this data. We observed that one of the ROIs for a patient in KL divergence skewed the distance to very high range while computing the kernel matrix. Hence, we removed that patient for KL divergence. The results shown here are for 14 patients using KL divergence and 15 patients in case of Bhattacharya and cepstral distance. The patient removed was marked as 4 by the clinicians.

5.5 Result analysis on refined ROI

5.5.1 Clustering results

The following are the conclusions arrived using Bhattacharya distance metrics and KL divergence to compute the kernel matrix.

The codword distribution across patients is very uneven on the refined ROI. For example, patient '1035' has few code words that appeared multiple times, but few codewords (almost 150 codewords) did not appear at all. We observed similar case with many patients. Because of this skewed distribution, we observed that there are no common codewords that appeared between two patients that are clustered together or clustered differently. The PCA variance of ROI data represented by the code word distribution tailed off linearly. It reveals that these data points are independent of each other. This is because the number of code words we chose (500) might be too many for 25 ROIs. Due to this reason, the distribution of code words is highly skewed.

1. Patients 1027 and 1023 are clustered together when computing kernel matrix using both KL divergence and Bhattacharya distance. For both these patients, cilia pixels are chosen for the analysis. Both are labeled as one by the clinicians. We can speculate that the CM patterns for both these patients might be similar and hence clustered together by both metrics.

2. Patients 7110 and 7096 are always grouped in different clusters using both KL and Bhattacharya distance. The pixels chosen from both these patients are mostly cilia and few from background. Patient 7110 is labeled two and patient 7096 is labeled as three by the clinicians. From above observations, we can probably say that these patients exhibit different CM patterns, hence grouped differently by both metrics.
3. Patients 7096 and 1035 are clustered in different groups in all cases when kernel matrix is computed using KL and Bhattacharya distance. The pixels chosen for patient 7096 contain mostly cilia along with few pixels from the background. Whereas pixels chosen for patient 1035 contains only cilia. Patient 1035 is marked as one and patient 7096 is marked as three by the clinicians. We can speculate that these two patients exhibit different CM patterns, hence clustered differently.
4. Patients 1013 and 1031 are clustered into the same group in most cases when Bhattacharya distance is used to compute the kernel. The ROIs for patient 1013 contain mostly cilia and few pixels from the cell body. Whereas the ROIs chosen for patient 1031 are mostly cilia and very few of them are from background of the cilia. Both of these patients are labeled as one by the clinicians. Since the data, we analyzed contains cell body and background we cannot conclude that these patients exhibit similar CM patterns.
5. Patients 7110 and 1035 are clustered together in most cases when Bhattacharya distance is used to compute the kernel. Pixels chosen for patient 1035 contain cilia. But, pixels chosen for patient 7110 contain both cilia and background. Patient 7110 is labeled as two and patient 1035 is labeled as one by the clinicians. We cannot speculate if these patients exhibit similar CM patterns as the pixels used in our analysis include few background pixels.

6. Patients 7110 and 7157 are clustered together using KL divergence kernel. Pixels chosen for patient 7110 contain both cilia and background, whereas pixels chosen for patient 7157 contain cilia. Patient 7110 is labeled as two and patient 7157 is labeled as four by the clinicians. Some of the CM patterns for these patients may or may not be similar, since the data analyzed on patient 7110 contains noise.
7. Patients 1023 and 1026 are clustered together for KL Divergence kernel. Cilia pixels are chosen for both these patients. Both these patients are labeled as one by the clinicians. We can speculate that both these patients might exhibit similar CM patterns and hence clustered together.
8. Patients 7127 and 7096 are clustered differently in most cases when the kernel matrix is computed using KL divergence. The ROIs chosen for patient 7127 is cilia, but the data chosen for patient 7096 contains both cilia and background pixels. Both patients are labeled as three by the clinicians. We can speculate that these 2 patients might exhibit different CM patterns.
9. Patients 1027 and 1013 are grouped in different clusters using KL divergence. Pixels chosen for both these patients are cilia. Both these patients are marked as normal (one) by the clinicians. Since we analyzed only cilia data we can probably say that CM patterns these patients exhibit might be different. Hence clustered in different groups.

The following is the result analysis when the kernel matrix is computed using cepstral distance.

1. Patients 1013 and 7127 are clustered in same group in most of the cases. The pixels chosen for both these patients are cilia. Patient 1013 is marked as one and patient 7127 is

marked as three by the clinicians. There might be portion of cilia for patient 7127 that has similar motion pattern as patient 1013 which is why our pipeline grouped them together.

2. Patients 1023 and 1035 are clustered together in most cases. Pixels chosen for both these patients contains only cilia. Both these patients are marked as one by the clinicians. We can speculate that these patients exhibit similar CM. Hence our pipeline grouped them together.
3. Patients 1027 and 7110 are grouped together in most of the results. Pixels chosen for both these patients consists of mostly cilia along with few pixels from the background. Patient 7110 is marked as two and patient 1027 is marked as one by the clinicians. Since the data we analyzed contains noise, we cannot conclude that these patients have similar CM patterns.
4. Patients 1035 and 7096 are clustered in different groups. Pixels chosen for patient 1035 contain of only cilia and pixels chosen for patient 7096 contain mostly cilia along with background. Patient 1035 is labeled as one and patient 7096 is labeled as three by the clinicians. We cannot speculate if these patients exhibit different CM, since the data we analyzed contains background.
5. Cluster results show that patients 7127 and 1035 are grouped in different clusters in most cases. Pixels chosen for both these patients contain cilia. Patient 7127 is marked as three and patient 1035 are marked as one by the clinicians. Based on our observations we can speculate that these two patients have different CM patterns.

A patient marked as three can exhibit multiple CM patterns that are a mixture of normal CM, wavy motion, and incomplete motion. Hence, a patient can exhibit multiple combination of

motion patterns that can be clustered together with patients that are labeled differently. In a similar way, a patient can be grouped in a different cluster although labeled with same number by the clinicians.

From the above clustering result analysis, we found that each distance metric is picking something different. There are no patients that are always clustered together or clustered differently across all three-distance metrics.

5.5.2 Classification Results

The following are the classification results on the refined ROI data for 15 patients. Here we applied 3-fold cross validation and k=3 for k-NN classification technique.

Result analysis using Bhattacharya kernel

The standard deviation for TF representation for all three classifiers per embedding is the same and are in the range 0.09 to 0.14. This low standard deviation indicates that the data points are closer to the mean of the set. We believe that there are less or no outliers in the data set. Standard deviation of TFIDF representation is also low for both SVM and random forest, but relatively high in case of k -NN. Classification accuracy for SVM and random forest across all four embedding techniques are mostly consistent but there is a slight variation in case of k-NN. Table 6 indicates the percentage accuracy results on 15 patients using Bhattacharya distance.

	SVM	*Random Forest*	*k-NN*	*Bagging*	*Ada boost*
MDS TF	*53.3*	*53.3*	*46.6*	*46.6*	*40*

MDS TFIDF	*53.3*	*47.7*	*47.7*	*53.3*	*53.3*
Isomap TF	*55*	*55*	*67.2*	*61.6*	*40*
Isomap TFIDF	*61.6*	*53.3*	*61.6*	*38.8*	*30*
PCA TF	*55.5*	*55.5*	*55*	*53.3*	*53.3*
PCA TFIDF	*55*	*55*	*22.2*	*55*	*55*
LEM TF	*55.5*	*55.5*	*55.5*	*53.3*	*53.3*
LEM TFIDF	*53.3*	*53.3*	*40*	*53.3*	*53.3*

Table 6: Classification results obtained by computing kernel matrix using Bhattacharya distance on 15 patients.

Result analysis using KL divergence

SVM classifier has the lowest standard deviation compared to random forest and k-NN. K-NN tend to have the most deviation in the results. High standard deviation indicates that the data is unevenly spread across the mean and produces high variation in the results. Hence, k -NN accuracy varied a lot from 18.3% through 61.6%. Random forest classifier predicted very well with an accuracy of 81.6 in case of LEM TFIDF but in few cases it could only predict with 66.6 % accuracy. This is due to the variance in the distribution of the data. Table 7 shows the percentage accuracy results for all four embedding techniques using KL divergence.

	SVM	*Random Forest*	*KNN*	*Bagging*	*Ada boost*
MDS TF	*63.3*	*68.3*	*58.3*	*61.1*	*66.6*
MDS TFIDF	*70*	*73.3*	*21.6*	*65*	*58.3*
Isomap TF	*66.6*	*75*	*58.3*	*58.8*	*58.8*
Isomap TFIDF	*66.6*	*66.6*	*61.6*	*50*	*36.6*
PCA TF	*70.8*	*65.8*	*55.8*	*48.3*	*41.6*
PCA TFIDF	*63.3*	*66.6*	*18.3*	*53.3*	*47.7*
LEM TF	*66.6*	*80*	*60*	*58.3*	*58.3*
LEM TFIDF	*70.8*	*81.6*	*47.5*	*56.6*	*35*

Table 7: Classification results obtained by computing kernel matrix using KL divergence on 14 patients.

Result analysis using cepstral distance

Standard deviation for SVM and random forest classifier are in similar range, where as the standard deviation of k -NN classifier is relatively higher than other two classifiers. Hence the accuracy varies a lot in case of k-NN classification. Table 8 represents the percentage accuracy across four embedding techniques using three classifiers.

	SVM	*Random Forest*	*KNN*	*Bagging*	*Ada boost*
MDS TF	*70.8*	*60.8*	*47.5*	*53.3*	*53.3*
MDS TFIDF	*66.6*	*61.6*	*38.3*	*53.3*	*53.3*
Isomap TF	*66.6*	*56.6*	*60*	*60*	*26*
Isomap TFIDF	*63.3*	*53.3*	*48.3*	*55.5*	*44.4*
PCA TF	*66.6*	*66.6*	*56.6*	*53.3*	*46.6*
PCA TFIDF	*66.6*	*66.6*	*65*	*53.3*	*53.3*
LEM TF	*70.8*	*65.8*	*55.8*	*53.3*	*53.3*
LEM TFIDF	*63.3*	*58.3*	*58.3*	*53.3*	*53.3*

Table 8: Classification results obtained by computing kernel matrix using cepstral distance on 15 patients.

5.6 Challenges

We faced a few technical and clinical challenges during our research. There is no clear understanding of different CM subtypes that exist and categorizing these subtypes is a very hard problem. Since all the clinicians do not agree upon the number of subtypes and the patterns they belong to, picking the number of clusters to categorize patients is a real challenge. Here we chose 4 clusters since all the clinicians agree that there are at least 4 CM subtypes that exists.

The ROIs we used here are manually chosen by the clinicians by drawing a patch around the cilia. But, these ROIs contain background and cell body of cilia along with the cilia. Also, there are some practical problems in the cilia videos as few videos are taken at different angles, at different frames per second and few videos are not clear enough to capture the complete motion. Preprocessing the ROI and choosing only cilia is a real challenge since the accuracy of the results is highly dependent on the data we chose. There is no automatic process to pick the cilia, finding the proper technique to pick cilia is challenging. Here we considered pixels with high intensity variation as cilia and performed our analysis.

AR parameters which are best for DT analysis in theory should work well but because of noisy input we had to look for other techniques. Martin distance, which depends on the AR parameters might not be a good option. Finding other distance metrics that do not use AR parameters and which capture non-linearity of the data is really challenging. After 2 months of research we found that time series analysis and cepstral analysis are suitable. Hence, we used KL divergence, Bhattacharya distance and cepstral distance.

Since we are working on pixel data, each ROI contains 100-pixel data across 200 frames. We have hundreds of such ROIs. Running these huge data sets consumed lot of time and resources. Finding optimal ways to run this data across multiple clusters and writing code to use optimal memory resources is challenging. The clusters crashed several times due to memory leaks and running out of CPU resources. Hence, we used profiling techniques to find parts of code causing this and optimized the code.

I was not a python programmer before this research project, hence learning a new language and writing efficient code was a challenge.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this thesis, we have provided algorithms and techniques which help in representing CM subtypes. We tried to group the patients based on the similarity and dissimilarity of CM patterns they exhibit. First, we identified the algorithms that can capture the non-linearity of the data and metrics that are invariant to transformations. We developed a novel framework that represents the patients using a bag of dynamical systems. This led to exploring different CM patterns a patient exhibits. We used unsupervised clustering techniques to bind the patients with similar CM patterns and separate the patients with dissimilar motion patterns. We also used classification techniques to explore relation between the degree of abnormality and CM subtypes. We explained detailed analysis of the results obtained and provided pseudocode in Appendix section. We also provided several challenges we faced during this analysis and the future work needed to improve up on our analysis.

Our analysis and investigation serves as a stepping-stone for finding CM subtypes. In the long run this can help the clinicians in early diagnosis and intervention of ciliopathies, which helps in implementing therapies that aims to cure a patient.

6.2 Future work

Future work includes finding pixel selection strategy that automatically selects only the data containing cilia by eliminating the noise. This helps in improving the accuracy of the techniques we follow.

Another area of improvement is scaling our analysis to work on more patients with more ROIs. This helps in analytical understanding of grouping the patients with similar CM on a larger scale. We envision that our current analysis of representing patients as a mixture of CM patterns serves as a basis to understand the CM subtypes in future.

APPENDICES

A Elemental components of optical flow

Consider two spatially nearby image points $\overrightarrow{r_1} = \vec{r}$ and $\overrightarrow{r_2} = \vec{r} + \delta\vec{r}$ along a cilium. The vector $\delta\vec{r} = \overrightarrow{r_2} - \overrightarrow{r_1}$ gives their relative position. We assume that the points move according to their optical flow velocities $\overrightarrow{f_1} = \vec{f} = (u, v)^T$ and $\overrightarrow{f_2} = \vec{f} + \delta\vec{f}$ and after a small-time interval δt they are at locations $\overrightarrow{r_1}' = \overrightarrow{r_1} + \overrightarrow{f_1}\delta t$ and $\overrightarrow{r_2}' = \overrightarrow{r_2} + \overrightarrow{f_2}\delta t$. It follows that

$$\overrightarrow{r_2}' - \overrightarrow{r_1}' = (\overrightarrow{r_2} - \overrightarrow{r_1}) + (\overrightarrow{f_2} - \overrightarrow{f_1})\,\delta t\,,$$

$$\delta\overrightarrow{r'} = \delta\vec{r} + \delta\vec{f}\delta t$$

Given the spatial closeness of 2 points $\overrightarrow{r_1}$ and $\overrightarrow{r_2}$, the optical flow vectors $\overrightarrow{f_1}$ and $\overrightarrow{f_2}$ can be related by Taylor series expansion that uses first order differentials of the optical flow:

$$\overrightarrow{f_2} \approx \overrightarrow{f_1} + \frac{\delta\overrightarrow{f_1}}{\delta\vec{r}}\,\delta\vec{r} + \cdots,$$

$$\overrightarrow{f_2} \approx \overrightarrow{f_1} + \begin{pmatrix} u_x & u_y \\ v_x & v_y \end{pmatrix}\delta\vec{r} + \cdots,$$

Where (u_x, u_y, v_x, v_y) are elements of spatial derivative of optical flow i.e. flow gradient $\frac{\delta\vec{f}}{\delta\vec{r}}$. Decomposing the flow gradient further gives scaling(divergence), shearing(deformation) and rotation (curl) components. These are the scalar quantities defines as

$$\text{Rot}\ \vec{f} = v_x - u_y$$

$$\text{Div}\ \vec{f} = u_x + v_y$$

$$\text{Def}\ \vec{f}\cos(2\mu) = u_x - v_y$$

$$\text{Def}\ \vec{f}\sin(2\mu) = u_y + v_x$$

Where μ is the angle of maximum distortion. The quantities Div $\vec{f}$, Rot $\vec{f}$, (Def $\vec{f}$) cos(2μ), (Def $\vec{f}$) sin(2μ) forms a linear space and provide an equivalent representation of flow gradient $\frac{\delta \vec{f}}{\delta \vec{r}}$. The quantities Def $\vec{f}$, Div $\vec{f}$ and rot $\vec{f}$ are the elemental components derived from optical flow and are also called differential invariants since they are independent of the coordinate system used to measure the flow.

Rotation

The most salient features of CM are sweeping forward and backward strokes. Curl or rotation captures the local rotation of the cilia with angular velocity ½ rot $\vec{f}$. Curl is orthogonal to divergence and is invariant to the orientation of cilia in image plane.

B Pseudocode of pipeline

This section explains the pseudocode used in our analysis.

```
Pseudocode 1: Pick top pixels with high intensity
Input: Rotation data with list of ROI names and their corresponding ROIs(n), each in the form of frames(f)*height(h)*width(w) ;
number of pixels with high standard deviation to keep(num_of_pixels);
maximum frequency per video to consider(T)
Output: List of ROI names and top pixels with high standard deviation from each ROI as a matrix of dimensions (n* num_of_pixels) x T

function Pick_Top_Pixels( n, top_sigma, T)
        Initialize the top_Pixels to empty set
        for each ROI in list of rotation Data
                Get only the top frames with in T
                Calculate the standard deviation for the above top frame data
                Sort these pixels with decreasing order of standard deviation
                Pick the top num_of_pixels pixels from the sorted pixels
                Assign the pixel values of these top num_of_pixels to top_Pixels by stacking them in a list along with
corresponding ROI_names
return top_pixels
```

Figure 29: Pseudocode to preprocess rotation data. This helps in picking pixels with high magnitude variation across time.

Pseudocode 2: Compute pairwise kernel matrix using Cepstral distance.

Input: x- List of ROI names along with the corresponding top 'p' pixels per ROI each across max_frames i.e. each row is in the form of [roi_name, vector of T features] and n * p(say N) such lists.

Output: A pairwise symmetric kernel matrix which contains the pair wise cepstral distance between each other of shape N x N.

function Cepstral_distance (x)

Initialize Cepstral_kernel to zeros with matrix of shape (N x N)

Compute the cepstrum of time series for each pixel using $(\hat{x_t})_{t=1}^{T} = IDFT(\ln(DFT\left((x_t)_{t=1}^{T}\right))$

for each index i in N

for each index j in N

Cepstral_kernel[i][j] = $\sum_{t=1}^{n} ||\hat{x_t}[i] - \hat{x_t}[j]||$

return Cepstral_kernel

Figure 30: Compute pairwise kernel matrix using cepstral distance between pixels.

Pseudocode 3: Compute pairwise kernel matrix using KL divergence distance.

Input: x- List of ROI names along with the corresponding top 'p' pixels per ROI each across max_frames i.e. each row is in the form of [roi_name, vector of T features] and n * p(say N) such lists; H – window size to compute the spectrum.

Output: A pairwise symmetric kernel matrix which contains the pair wise KL distance between each other of shape N x N.

function KL_divergence(x, H)

Initialize KL_kernel to zeros with matrix of shape (N x N)

Compute the Spectrum of time series for each pixel using $F(V_k) = \sum_{i=k-H}^{k+H} G_k$ where

G_k is the periodogram computed using $G_k = f^1_k f^2_k$ where

fk is the component wise Fast Fourier Transformation computed using f (i,:) = FFT(X(i,:)), and set f_k = f(:,k)

for each index i in N

for each index j in N

KL_kernel[i][j] = $\sum_{0<vk<\frac{1}{2}} [trace\{F_1(V_k)\,F_2{}^{-1}(V_k)\} - \ln\frac{|F_1(V_k)|}{|F_2(V_k)|} - N]$

return KL_kernel

Figure 31: Compute pairwise kernel matrix using KL divergence between pixels.

Pseudocode 4: Compute pairwise kernel matrix using Bhattacharya distance.

Input: x- List of ROI names along with the corresponding top 'p' pixels per ROI each across max_frames i.e. each row is in the form of [roi_name, vector of T features] and n * p(say N) such lists;

α – parameter and is equal to 05. for bhattacharya distance;

H – window size to compute the spectrum.

Output: A pairwise symmetric kernel matrix which contains the pair wise Bhattacharya distance between each other of shape N x N.

function Bhattacharya_distance(x, H, α)

Initialize Bhattacharya_kernel to zeros with matrix of shape (N x N)

Compute the Spectrum of time series for each pixel using $F(V_k) = \sum_{i=k-H}^{k+H} G_k$ where

G_k is the periodogram computed using $G_k = f^1_k f^2_k$ where

fk is the component wise Fast Fourier Transformation computed using f (i,:) = FFT(X(i,:)), and set f_k = f(:,k)

for each index i in N

for each index j in N

$$\text{Bhattacharya\_kernel} = 1/2 \sum_{0<vk<\frac{1}{2}} [\ln \frac{|\alpha F_1(V_k)+(1-\alpha)F_2(V_k)|}{|F_2(V_k)|} - \alpha \ln \frac{|F_1(V_k)|}{|F_2(V_k)|}]$$

return Bhattacharya_kernel

Figure 32: Compute pairwise kernel matrix using Bhattacharya distance between pixels.

Pseudocode 5: To compute code word and membership of each data point

Input: Pairwise kernel computed using Cepstral(or KL divergence or Bhattacharya distance), ker- of shape N x N ; Number of codewords to form (k); Top pixels data(x)

Output : List of membership to the k codewords of length N(number_of_ROI * top_pixels). We get 4 such lists for MDS, Isomap, PCA and LEM

MDS – Multi Dimensional Scaling

PCA – Principal Component Analysis

LEM – Laplacian Eigen Maps

function codeword (ker, k, x):

Compute manifold on ker using MDS, Isomap, PCA, LEM by reducing the dimensions to 2.

1 Compute k means clustering on the manifold data and obtain k cluster centers

2 Obtain the k indices of ker which are closer to these cluster centers, using Euclidean distance.

3 Get the corresponding k data points from x(original top pixel data) based on the above indices. This is code book.

4 Compute the membership of each data point of x to code word by considering the minimum distance between them.

Compute all the above steps from 1 trough 4 for all 4 manifold data.(MDS, Isomap, PCA and LEM)

return membership_MDS, membership_Isomap, membership_PCA, memebership_LEM

Note: While computing the membership of each data point, we used corresponding distance metric of the input ker. If ker is cepstral, we used cepstral distance to compute distance from each data point to the codeword and assigned the minimum distance code word to that point. Similarly if ker is from kl divergence, we used kl distance.

Figure 33: Compute code word from the kernel matrix. Then compute the membership of original data with respect to each code word.

Pseudocode 6: To compute patient weight representation using weight vectors.

Input : List of membership to the k codewords of length N(number_of_ROI * top_pixels). We get 4 such lists for MDS, Isomap, PCA and LEM

Output: List of k weight vectors for each patient along with patient name. We obtain 2 separate lists one computed using Term frequency and other using Term frequency Inverse document frequency

Function patient_weight_representation(membership_MDS, membership_Isomap, membership_PCA, membership_LEM):

Calculate the TF for each patient by computing w_{ik} = Number of times codeword k occurs in i^{th} patient/ Total number of code words in i^{th} patient.

Calculate TFIDF for each patient by computing w_{ik} = TF for i^{th} patient * log(total number of video sequences/ Total number of patients in which code word i occurs.

Normalize the TF weight vector using L1 norm to become a histogram.

return normalized_tf_weightVector, TFIDF_weightVector

Figure 34: Compute patient weight representation using TF and TF -IDF

Pseudocode 7: To compute similarity matrix

Input: List of k weight vectors for each patient.

Output : n x n similarity matrix, where n is the number of patients

function similarity matrix(weight_vectors):

Initialize n*n matrix with zeros for chi square kernel.

Initialize n*n matrix with zeros for cosine kernel.

Compute chi square distance using $d_{\chi^2}(W_1, W_2) = \frac{1}{2}\sum_{i=1}^{K}\frac{|w_{1i} - w_{2i}|}{w_{1i}+w_{2i}}$

Compute cosine similarity using $d_{cosine}(W_1, W_2) = \frac{W_1 W^T{}_2}{||W_1||||W_2||}$

return chisquare_kernel, cosine_kernel

Figure 35: Compute distance between weight vectors.

REFERENCES

1. Jason M. Brown and George B. Witman, “Cilia and Diseases.”, bioscience (2014) Volume 64, Issue 12: 1126-1137
2. Satir, Peter, and Søren T. Christensen. “Structure and Function of Mammalian Cilia.” Histochemistry and Cell Biology 129.6 (2008): 687–693.
3. Lee, Ji E., and Joseph G. Gleeson. “Cilia in the Nervous System: Linking Cilia Function and Neurodevelopmental Disorders.” Current opinion in neurology 24.2 (2011): 98–105.
4. Tilley A.E., Walters M.S., Shaykhiev R., Crystal R.G, “Cilia dysfunction in lung disease.” Annual Review of Physiology 2015; Volume:77: 379–406.
5. Enuka Y, Hanukoglu I, Edelheit O, Vaknine H, Hanukoglu A (Mar 2012). "Epithelial sodium channels (ENaC) are uniformly distributed on motile cilia in the oviduct and the respiratory airways." Histochemistry and Cell Biology. 137 (3): 339–53.
6. Hanukoglu I, Hanukoglu A (Jan 2016). "Epithelial sodium channel (ENaC) family: Phylogeny, structure-function, tissue distribution, and associated inherited diseases." Gene. 579 (2): 95–132.
7. O’callaghan, Christopher, et al. "Diagnosing primary ciliary dyskinesia." (2007): 656-657.
8. M. W. Leigh, J. E. Pittman, J. L. Carson, T. W. Ferkol, S. D. Dell, S. D. Davis, M. R. Knowles, M. A. Zariwala, “Clinical and genetic aspects of primary ciliary dyskinesia/Kartagener syndrome.” Genet. Med. 11, 473–487 (2009).

9. Daniels, M. Leigh Anne, and Peadar G. Noone. “Genetics, Diagnosis, and Future Treatment Strategies for Primary Ciliary Dyskinesia.” Expert opinion on orphan drugs 3.1 (2015): 31–44.
10. M. Swisher, R. Jonas, X. Tian, E. S. Lee, C. W. Lo, L. Leatherbury, “Increased post-operative and respiratory complications in patients with congenital heart disease associated with heterotaxy.” J. Thorac. Cardiovasc. Surg. 141, 637–644.e3 (2011).
11. B. Thomas, A. Rutman, R. A. Hirst, P. Haldar, A. J. Wardlaw, J. Bankart, C. E. Brightling, C. O’Callaghan, “High prevalence of respiratory ciliary dysfunction in congenital heart disease patients with heterotaxy.” Circulation 125, 2232–2242 (2012).
12. B. Harden, X. Tian, R. Giese, N. Nakhleh, S. Kureshi, R. Francis, S. Hanumanthaiah, Y. Li, M. Swisher, K. Kuehl, I. Sami, K. Olivier, R. Jonas, C. W. Lo, L. Leatherbury, “Increased postoperative respiratory complications in heterotaxy congenital heart disease patients with respiratory ciliary dysfunction.” J. Thorac. Cardiovasc. Surg. 147, 1291–1298.e2 (2014).
13. P. K. Yiallouros, P. Kouis, N. Middleton, M. Nearchou, T. Adamidi, A. Georgiou, A. Eleftheriou, P. Ioannou, A. Hadjisavvas, K. Kyriacou, “Clinical features of primary ciliary dyskinesia in Cyprus with emphasis on lobotomized patients.” Respir. Med. 109, 347–356 (2015).
14. W. A. Stannard, M. A. Chilvers, A. R. Rutman, C. D. Williams, C. O’Callaghan, “Diagnostic testing of patients suspected of primary ciliary dyskinesia.” Am. J. Respir. Crit. Care Med. 181,307–314 (2010).

15. S. Dimova, F. Maes, M. E. Brewster, M. Jorissen, M. Noppe, P. Augustijns, “High-speed digital imaging method for ciliary beat frequency measurement.” J. Pharm. Pharmacol. 57, 521–526 (2005).
16. M. A. K. Olm, J. E. Kögler Jr., M. Macchione, A. Shoemark, P. H. N. Saldiva, J. C. Rodrigues, “Primary ciliary dyskinesia: Evaluation using cilia beat frequency assessment via spectral analysis of digital microscopy images.” J. Appl. Physiol. 111,295–302 (2011).
17. G. Mantovani, M. Pifferi, G. Vozzi, “Automated software for analysis of ciliary beat frequency and meta-chronal wave orientation in primary ciliary dyskinesia.” Eur. Arch. Otorhinolaryngol. 267, 897–902 (2010).
18. C. O’Callaghan, K. Sikand, M. Chilvers, “Analysis of ependymal ciliary beat pattern and beat frequency using high speed imaging: Comparison with the photomultiplier and photodiode methods.” Cilia 1, 8 (2012).
19. B. Thomas, A. Rutman, C. O’Callaghan, “Disrupted ciliated epithelium shows slower ciliary beat frequency and increased dyskinesia.” Eur. Respir. J. 34, 401–404 (2009).
20. C. M. Smith, R. A. Hirst, M. J. Bankart, D. W. Jones, A. J. Easton, P. W. Andrew, C.O’Callaghan, “Cooling of cilia allows functional analysis of the beat pattern for diagnostic testing.” Chest 140, 186–190 (2011).
21. C. Clary-Meinesz, J. Cosson, P. Huitorel, B. Blaive, “Temperature effect on the ciliary beat frequency of human nasal and tracheal ciliated cells.” Biol. Cell 76, 335–338 (1992).
22. M. Salathe, “Regulation of mammalian ciliary beating.” Annu. Rev. Physiol. 69, 401–422 (2007).

23. J. Raidt, J. Wallmeier, R. Hjeij, J. G. Onnebrink, P. Pennekamp, N. T. Loges, H. Olbrich, K. Häffner, G. W. Dougherty, H. Omran, C. Werner, “Ciliary beat pattern and frequency in genetic variants of primary ciliary dyskinesia.” Eur. Respir. J. 44,1579–1588 (2014).
24. Shannon P. Quinn, Maliha J. Zahid, John R. Durkin, Richard J. Francis, Cecilia W. Lo and S. Chakra Chennubhotla, “Automated Identification of abnormal respiratory ciliary motion in nasal biopsies”, Science Translational Medicine, 2015.
25. G. Doretto, A. Chiuso, Y. N. Wu,S. Soatto, “Dynamic textures.” Int. J. Comp. Vis. 51,91–109 (2003).
26. G. Zhao, M. Pietikainen, “Dynamic texture recognition using local binary patterns with an application to facial expressions.” IEEE Trans. Pattern Anal. Mach. Intell. 29, 915–928 (2007).
27. Z. Lu, W. Xie, J. Pei, J. Huang, “Dynamic texture recognition by spatio-temporal multiresolution histograms”, Seventh IEEE Workshop on Application of Computer Vision, WACV/MOTIONS’05 (Volume 2), Breckenridge, CO, 5 to 7 January 2005.
28. J. Chen, G. Zhao, M. Salo, E. Rahtu, M. Pietikamen, “Automatic dynamic texture segmentation using local descriptors and optical flow.” IEEE Trans. Image Process. 22, 326–339 (2013).
29. D. Sun, S. Roth, M. J. Black, “Secrets of optical flow estimation and their principles, Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.” (CVPR 2010), San Francisco, CA, 13 to 18 June 2010
30. Doretto, Gianfranco, et al. "Dynamic Texture Segmentation." *ICCV*. Vol. 2. 2003.

31. P. Saisan, G. Doretto, Y. N. Wu, S. Soatto, “Dynamic texture recognition”, Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Volume 2), 2001.
32. J. Huang, X. Huang, D. Metaxas, L. Axel, “Dynamic texture based heart localization and segmentation in 4-D cardiac images”, 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Arlington, VA, 12 to 15 April 2007.
33. N. Brieu, N. Navab, J. Serbanovic-Canic, W. H. Ouwehand, D. L. Stemple, A. Cvejicb, M. Grohera, “Image-based characterization of thrombus formation in time-lapse DIC microscopy.” Med. Image Anal. 16, 915–931 (2012).
34. Boon, Mieke et al. “Primary Ciliary Dyskinesia: Critical Evaluation of Clinical Symptoms and Diagnosis in Patients with Normal and Abnormal Ultrastructure.” Orphanet Journal of Rare Diseases 9 (2014): 11.
35. Avinash Ravichandran, Rizwan Chaudhry and rene Vedal , “View-Invariant Dynamic Texture Recognition using a Bag of Dynamical Systems”, Centre for Imaging Science, Johns Hopkins University, Baltimore
36. Woolfe F et al, “Shift-invariant dynamic texture recognition,” in Computer Vision–ECCV 2006, pp.549–562. Springer, 2006.
37. L.J.P. van der Maaten and G.E. Hinton. “Visualizing High-Dimensional Data Using t-SNE.” Journal of Machine Learning Research 9(Nov):2579-2605, 2008.
38. M. Hyndman, A. D. Jepson, D. J. Fleet, Higher-order autoregressive models for dynamic textures, British Machine Vision Conference, Warwick, 2007.
39. Martin RJ, “A metric for arma processes,” Signal Processing, IEEE Transactions on, vol. 48, no. 4, pp. 1164–1170, 2000.

40. De Cock K et al, “Subspace angles between arma models,” Systems & Control Letters, vol. 46, no. 4, pp. 265–270, 2002.
41. Puybareau, E., et al. "Automating the measurement of physiological parameters: a case study in the image analysis of cilia motion." Image Processing (ICIP), 2016 IEEE International Conference on. IEEE, 2016.
42. Anneloes Dummer, Christian Poelma, Marco C. DeRuiter, Marie-José T. H. Goumans, Beerend P. Hierck, “Measuring the primary cilium length: improved method for unbiased high-throughput analysis.” Cilia, 2016, Volume 5.
43. Laura J. Martin, MD, MPH, ABIM Board Certified in Internal Medicine and Hospice and Palliative Medicine, “https://medlineplus.gov/ency/imagepages/19533.htm”
44. M. A. Chilvers, M. McKean, A. Rutman, B. S. Myint, M. Silverman, C. O’Callaghan, “The effects of coronavirus on human nasal ciliated respiratory epithelium.” Eur. Respir. J. 18, 965–970 (2001).
45. B. Thomas, A. Rutman, R. A. Hirst, P. Haldar, A. J. Wardlaw, J. Bankart, C. E. Brightling, C. O’Callaghan, “Ciliary dysfunction and ultrastructural abnormalities are features of severe asthma.” J. Allergy Clin. Immunol. 126, 722–729.e2 (2010).
46. C. O’Callaghan, M. Chilvers, C. Hogg, A. Bush, J. Lucas, “Diagnosing primary ciliary dyskinesia.” Thorax 62, 656–657 (2007)
47. Stillwell, Paul C., Eric P. Wartchow, and Scott D. Sagel. “Primary Ciliary Dyskinesia in Children: A Review for Pediatricians, Allergists, and Pediatric Pulmonologists.” Pediatric Allergy, Immunology, and Pulmonology 24.4 (2011): 191–196.
48. Hughes, Daniel. “Primary Ciliary Dyskinesia.” Paediatrics & Child Health 13.8 (2008): 672–674.

49. “Ciliopathies: an expanding disease spectrum”, Aoife M. Waters and Philip L. Beales.
50. Aylsworth AS, “Clinical aspects of defects in the determination of laterality,” American Journal of Medical Genetics, vol. 101, no. 4, pp. 345–355, 2001.
51. Garrod AS et al, “Airway ciliary dysfunction and sinopulmonary symptoms in patients with congenital heart disease,” Annals of the American Thoracic Society, vol. 11, no. 9, pp. 1426–1432, 2014.
52. I. Narang, R. Ersu, N. Wilson, and A. Bush, “Nitric oxide in chronic airway inflammation in children: diagnostic use and pathophysiological significance,” Thorax, vol. 57, no. 7, p. 586, 2002.
53. N. Nakhleh, M. Swisher, R. Francis, R. Giese, B. Chatterjee, P. Connelly, I. Sami, K. Kuehl, K. Olivier, R. Jonas et al., “diag.” American Journal of Respiratory and Critical Care Medicine, vol. 179, no. 1 MeetingAbstracts, p. A2213, 2009.
54. A. Bush, P. Cole, M. Hariri, I. Mackay, G. Phillips, C. O’Callaghan, R. Wilson, and J. Warner, “Primary ciliary dyskinesia: diagnosis and standards of care,” European Respiratory Journal, vol. 12, no. 4, p. 982, 1998
55. C. Rayner, A. Rutman, A. Dewar, M. Greenstone, P. Cole, and R. Wilson, “Ciliary disorientation alone as a cause of primary ciliary dyskinesia syndrome,” American Journal of Respiratory and Critical Care Medicine, vol. 153, no. 3, p. 1123, 1996.
56. E. Escudier, M. Couprie, B. Duriez, F. Roudot-Thoraval, M. Millepied, V. Pruliere-Escabasse, L. Labatte, and A. Coste, “Computer-assisted analysis helps detect inner dynein arm abnormalities,” American Journal of Respiratory and Critical Care Medicine, vol. 166, no. 9, p. 1257, 2002

57. P. Noone, M. Leigh, A. Sannuti, S. Minnix, J. Carson, M. Hazucha, M. Zariwala, and M. Knowles, “Primary ciliary dyskinesia: diagnostic and phenotypic features,” American Journal of Respiratory and Critical Care Medicine, vol. 169, no. 4, p. 459, 2004.
58. M. Zariwala, M. Knowles, and H. Omran, “Genetic defects in ciliary structure and function,” Physiology, vol. 69, no. 1, p. 423, 2007.
59. M. Chilvers, A. Rutman, and C. O’Callaghan, “Ciliary beat pattern is associated with specific ultrastructural defects in primary ciliary dyskinesia,” Journal of Allergy and Clinical Immunology, vol. 112, no. 3, pp. 518–524, 2003.
60. M. Swisher, R. Jonas, X. Tian, E. S. Lee, C. W. Lo, and L. Leatherbury, “Increased postoperative and respiratory complications in patients with congenital heart disease associated with heterotaxy,” The Journal of Thoracic and Cardiovascular Surgery, vol. 141, no. 3, pp. 637–644, 2011.
61. M. Zahid, O. Khalifa, W. Devine, C. Yau, R. Francis, D. M. Lee, K. Tobita, P. Wearden, L. Leatherbury, S. Webber, and C. W. Lo, “Airway ciliary dysfunction in patients with transposition of the great arteries,” in Circulation, vol. 126, 2012, p. A15746.
62. http://docs.opencv.org/2.4/modules/video/doc/motion_analysis_and_object_tracking.html #calcopticalflowfarneback.
63. Shumway, R.H., Stoffer, D.S.: Time Series Analysis and its Applications. Springer (2000).
64. http://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html
65. S. Quinn, R. Francis, C. Lo, C. Chennubhotla, “Novel use of differential image velocity invariants to categorize ciliary motion defects”, Biomedical Sciences and Engineering Conference (BSEC), Knoxville, TN, 15 to 17 March 2011.

66. shttp://scikit-learn.org/stable/modules/manifold.html#t-sne

67. “Primary ciliary dyskinesia: when to suspect the diagnosis and how to confirm it.” Hogg, Claire. Paediatric Respiratory Reviews , Volume 10 , Issue 2 , 44 – 50.

68. http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html