Stream Sequence Mining for Human Activity Discovery

Parisa Rashidi

Biomedical Informatics Division,
Health and Engineering Center,
Northwestern University,
Chicago, IL, US

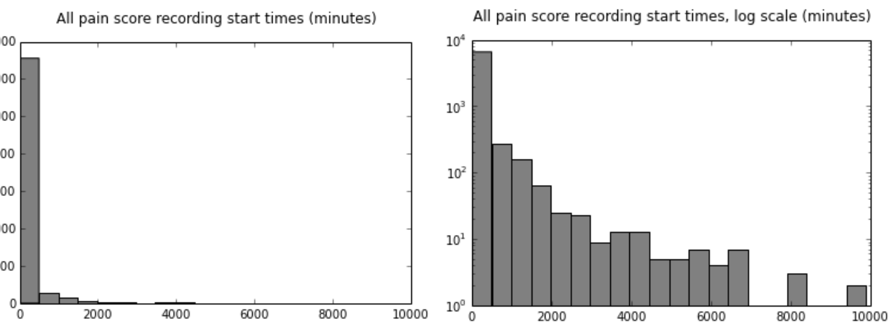
Keywords: Activity Discovery, Activity Recognition, Unsupervised Learning, Sequence Mining, Stream Mining

todo: re-save figures in pdf format

1. Methods

1.1. Standardizing start times

Most pain scores in the data file begin shortly after surgery, with about 80% starting within two hours post-operation. However, there is a non-trivial subset of patients whose pain score recordings started up to several days after their surgeries, as shown in Figure 1.



(a) Normal scale

(b) Log scale

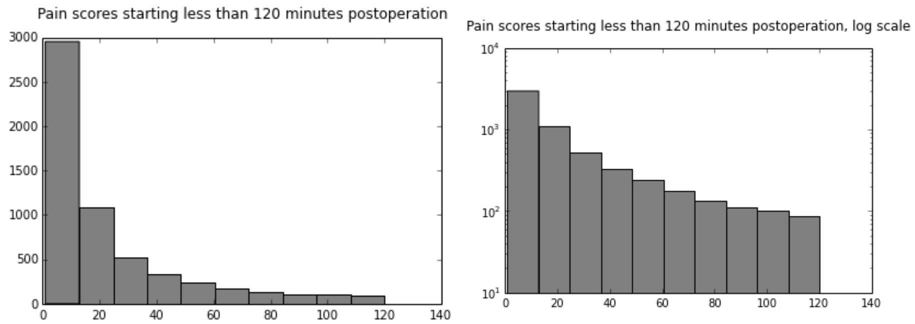
Figure 1: Histograms of all pain score start times in minutes, both normal and logarithmic scales

In the interest of analyzing pain profiles along a standard post-operation time interval, data was disregarded for those patients whose pain scores were

Email address: `parisa.rashidi@northwestern.edu` (Parisa Rashidi)

Preprint submitted to Book on Plan, Activity, and Intent Recognition *June 10, 2014*

recorded starting more than two hours after their procedure. This operation retained 80% of the original data set. Figure 2 shows the set of start times less than two hours post-operation.



(a) Normal scale

(b) Log scale

Figure 2: Histograms of all pain score start times less than 120 minutes, both normal and logarithmic scales

1.2. Interpolation

Let p refer to an arbitrary patient whose pain score records started less than two hours post-operation. Since pain scores were recorded at inconsistent intervals, it is necessary to standardize them to consistent intervals. Ten minutes was chosen as the standard interval between scores, and recorded scores were linearly interpolated and estimated for these periods. Figure 3 illustrates the transformation from sample recorded scores, (1), to interpolated scores, (2) (rounded to the hundredths place for display purposes).

$$recorded = (3\ minutes : 4, 24\ minutes : 8, 45\ minutes : 4, 100\ minutes : 2) \tag{1}$$

$$\begin{aligned} interpolated = (&3\ minutes : 4.0, 13\ minutes : 5.9, 23\ minutes : 7.8, \\ &33\ minutes : 6.3, 43\ minutes : 4.4, 53\ minutes : 3.7, \\ &63\ minutes : 3.3, 73\ minutes : 3.0, 83\ minutes : 2.6, \\ &93\ minutes : 2.0) \end{aligned} \tag{2}$$

Denote patient p's time series of 10-minute interpolated pain scores as $\tilde{S} = (s_1, ..., s_d)$, where the t'th item is the estimated pain score $10(t-1)$ minutes past the first recorded score, and $d = floor\left(\frac{max(recordingtime)}{10}\right)$. So s_1 is the first pain score recorded for that patient with subsequent elements estimated from linear interpolation, and d is the index of the last available data point estimated from interpolation without forecasting past the last recorded pain score. For the example interpolated values in (2), $\tilde{S} = (4.0, 5.9, 7.8, 6.3, 4.4, 3.7, 3.3, 3.0, 2.6, 2.3)$.

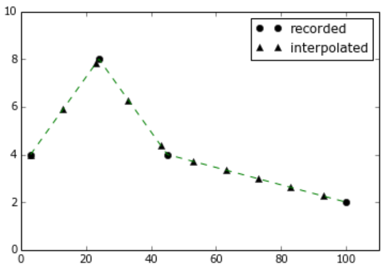


Figure 3: Illustration of interpolation procedure. Recorded pain scores are connected via a linear function of time from which interpolated scores are predicted at consistent, ten-minute intervals.

1.3. Calculating motif occurrences

Each motif of length l is represented by a vector of alphabet letter indices $(k_1, k_2, ..., k_l)$, 1-based such that motif adb would be $(1, 4, 2)$. Define a matrix of motif occurences M from a collection of motifs such that $M_{p,m}$ refers to the occurrence count of motif m within a given SAX sequence representing the post-operative pain scores for patient p (pain scores are linearly interpolated to ten minute increments). For any given motif of length l represented by up to β letters of the alphabet, its position in the matrix is defined by Equation (3).

$$m = 1 + \sum_{j=1}^{l} (k_j - 1) \times \beta^{l-j} \tag{3}$$

For example, given a motif such as adb, with $\beta = 4$ (that is, an alphabet letter selection of $\{a, b, c, d\}$), then $m = 14$. This has the effect of placing the motifs in alphabetical order within the vector $M_{p,\cdot}$. Note that the total number of possible motifs of length l with β alphabet letters is β^l. In our analysis, l was restricted to a value of 2, and values of β were examined along the set $\{2, 5, 10\}$.

1.4. Clustering

Define multiple clusters of patients $(C_{\kappa_1}, ..., C_{\kappa_n})$ such that $p \in C_{\kappa_i}$, iif criterion κ_i describes patient p. A given criterion κ may specify the patient's gender, age group, surgery type, etc. or include a set of multiple criteria. The relative importance $x_{C_\kappa,m}$ of motif m to a given cluster C_κ can therefore be calculated as follows:

$$x_{C_\kappa,m} = \frac{\sum\limits_{p \in C_\kappa} M_{p,m}}{\sum\limits_{p \in C_\kappa} \sum\limits_{m} M_{p,m}} \tag{4}$$

In other words, the relative importance is equal to the occurence count for motif m within the recoveries of all patients in the cluster divided by the total occurence count for all motifs within those recoveries.

Note that a patient could hypothetically be assigned to multiple clusters if, for example, the analysis looked at a feature whose value changed during the course of data collection (such as the patient started on a new medication during the course of his/her recovery). While this is potentially an area where the procedure could be improved, the data file does not appear to exhibit this behavior. Feature values were static for any given patient, and thus there would be no duplication between clusters.

1.5. Normalizing

In addition to comparing all motif importance values within the same cluster, the goal is also to compare individual motifs importance values across clusters. As such it is necessary to normalize $x_{C_\kappa,m}$ in order to elucidate subtle differences across all relevant cluster criteria. One simple and effective option is to use the standard score $z_{C_\kappa,m}$.

In a realistic analysis, the number of clusters is generally very small. For example, if $\kappa \in \{\textit{patient is taking an SSRI}, \textit{patient is not taking an SSRI}\}$, then the mean $\mu_{\cdot,m}$ and standard deviation $\sigma_{\cdot,m}$ of importance values for motif m across all clusters in the analysis only take into account two values for $x_{C_\kappa,m}$, hardly enough to calculate a useful standard score. Therefore, instead of normalizing values of $x_{\cdot,m}$ directly against each other, values of $x_{C_\kappa,m}$ were normalized against the random distribution $x^*_{C_\kappa,m}$, bootstrapped by taking a sufficiently large number of random samples of relative importance values. For any given cluster C_κ and motif m,

1. Let $|C_\kappa|$ be the number of patients in C_κ.
2. Take some arbitrarily large number n (say, 1000) of random samples $(C_{r_1}, ..., C_{r_n})$ so that each sample represents a cluster of $|C_\kappa|$ patients independently chosen from $\{\textit{all patients}\}$ with replacement and uniform probability of being chosen.
3. For each sample C_r, calculate the relative motif importance $x_{C_r,m}$. Refer to this new set, $\left\{x_{C_{r_1},m}, x_{C_{r_2},m}, ..., x_{C_{r_n},m}\right\}$, as $x^*_{C_\kappa,m}$. Ideally $x^*_{C_\kappa,m}$ will resemble a normal distribution. As figure 4 shows, this appears to be the case from the data set for example values of $n_r = 1000$, $|C_\kappa| = 100$, and $\beta = 5$. Although, this is not necessarily the case for every $x^*_{C_\kappa,m}$ when β is sufficiently large or $|C_\kappa|$ is sufficiently small. Figure 5 shows $x^*_{C_\kappa,m}$ histograms when $n_r = 1000$, $|C_\kappa| = 100$, and $\beta = 10$. Near two of the corners $x^*_{C_\kappa,m}$ values hover around 0. The reason is that instances of such drastic jumps in pain scores (j to a or a to j) are relatively rare. For practical purposes, the visual effect on the final icons as well as the effect on sorting the icons is negligible, so this small source of normalization error is acceptable. However, figure 6 shows $x^*_{C_\kappa,m}$ histograms when $n_r = 1000$, $|C_\kappa| = 2$, and $\beta = 10$. Data calculated for these parameters are clearly not normal, which is unsurprising given that the numerator in $x_{C_\kappa,m}$ expresses additive behavior over potentially rare events; that is, the sum of the number of times patients in a given cluster went from one normalized pain level to another. Therefore, in this analysis, which limited β to a maximum value of 10, only clusters which contained at least 100 patients ($|C_\kappa| \geq 100$) were considered. As it turns out, most interesting clusters

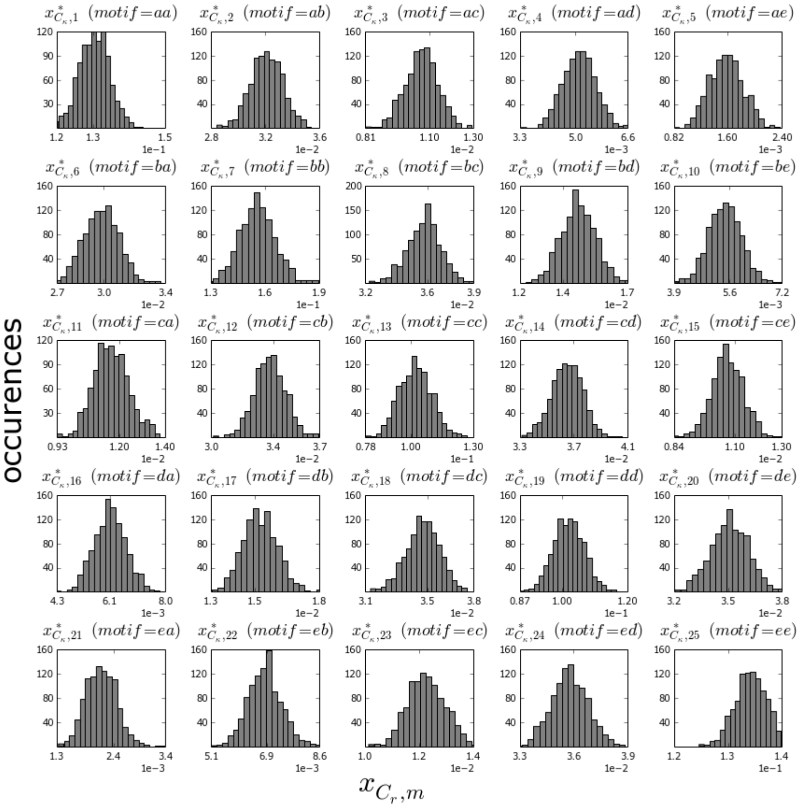


Figure 4: Histograms of resampled importance values for each motif, with parameters $n_r = 1000$, $|C_\kappa| = 100$, and $\beta = 5$. Demonstrates ideal normal behavior of the simulated importance value distributions.

have several hundred patients, so this limit is rarely an issue. In the future, however, other probability distributions may prove useful in analyzing these smaller clusters.

4. From $x^*_{C_\kappa,m}$, calculate the mean $\mu_{C_\kappa,m}$ and standard deviation $\sigma_{C_\kappa,m}$. The normalized value of $x_{C_\kappa,m}$ is therefore the standard score

$$z_{C_\kappa,m} = \frac{x_{C_\kappa,m} - \mu_{C_\kappa,m}}{\sigma_{C_\kappa,m}} \tag{5}$$

The vertical lines in figure 7 represent values for $z_{C_\kappa,m}$ plotted against histograms for standard scores of all values in $x^*_{C_\kappa,m}$; that is, equation (5) applied to all sampled points in $x^*_{C_\kappa,m}$ (creating a derived distribution, $z^*_{C_\kappa,m}$). Specifically, these data represent parameter values $\beta = 5$, $n_r = 1000$, and $\kappa =$"*patient underwent cardiovascular surgery*" (PrimaryCPTCodeCategory2 = *Cardiovascular*), a cluster which contains 582 patients ($|C_\kappa| = 582$).

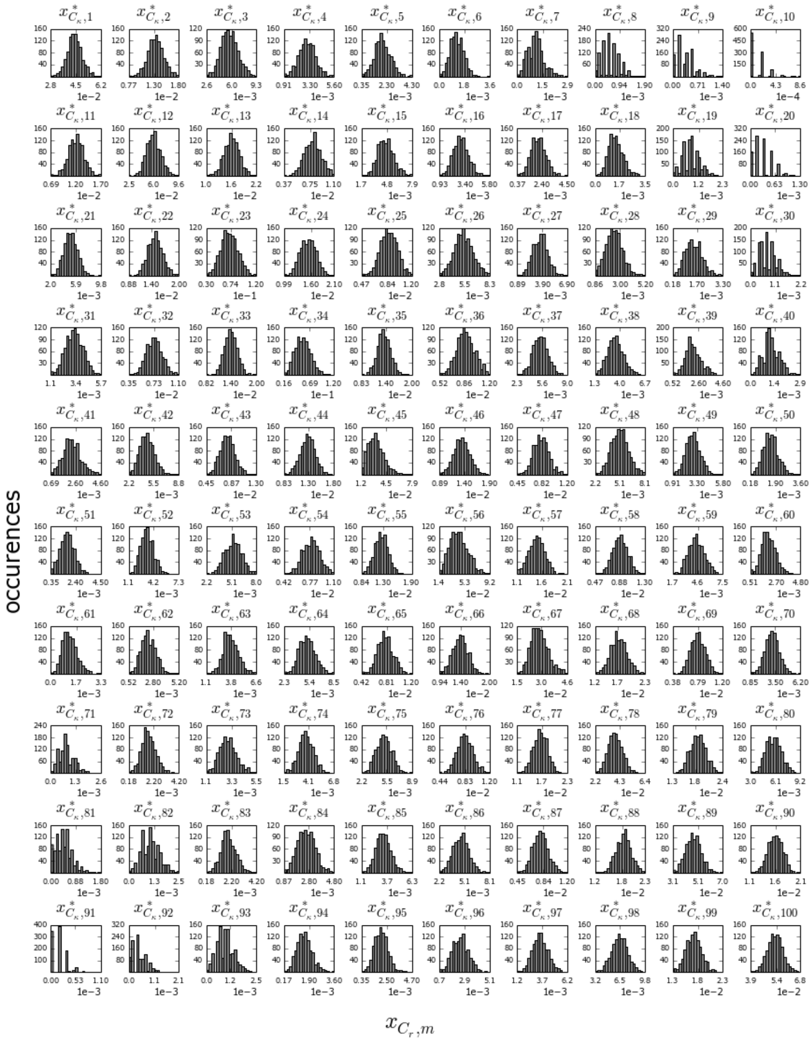


Figure 5: Histograms of resampled importance values for each motif, with parameters $n_r = 1000$, $|C_\kappa| = 100$, and $\beta = 10$. Demonstrates slight deviation from normal behavior around the upper-right and lower-left corners.

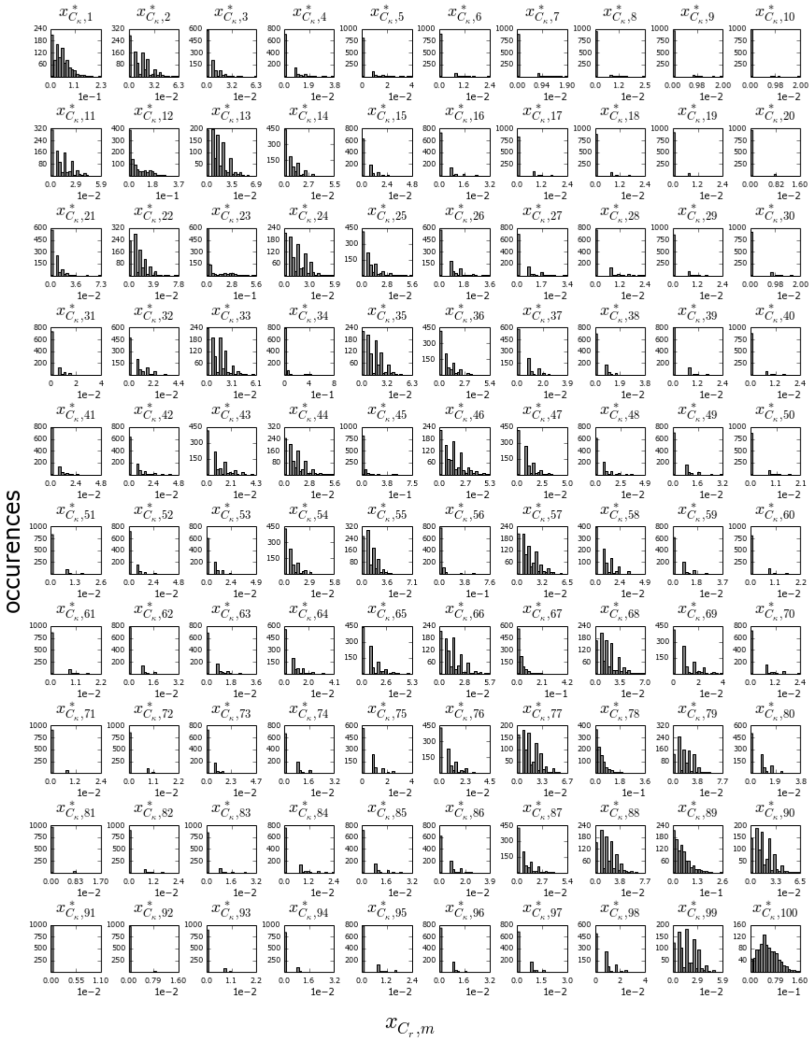


Figure 6: Histograms of resampled importance values for each motif, with parameters $n_r = 1000$, $|C_\kappa| = 2$, and $\beta = 10$. Demonstrates collapse of normality expectation and necessitates the restriction on cluster sizes of less than 100 patients.

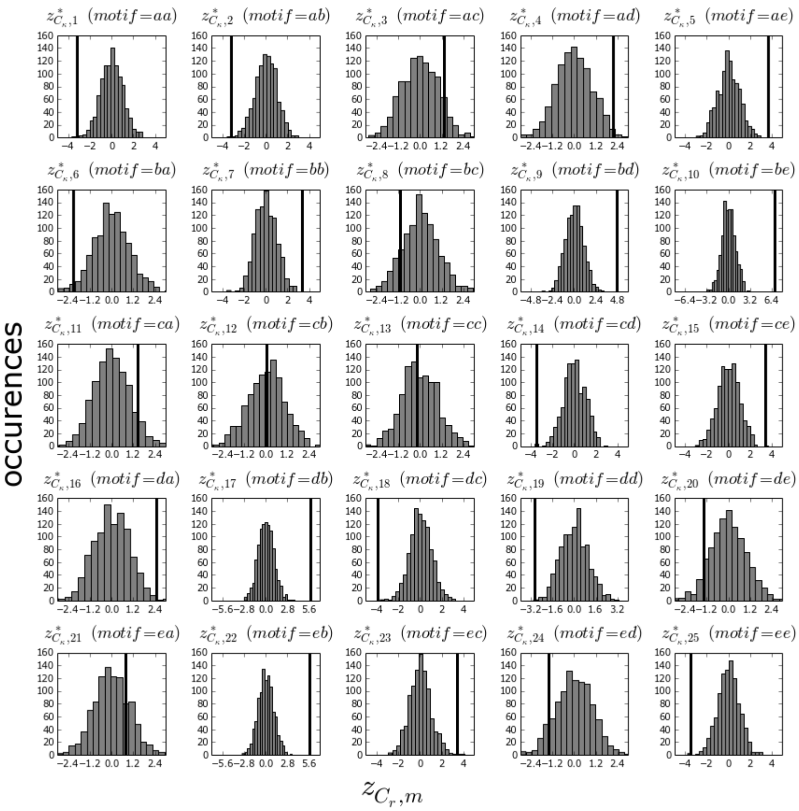


Figure 7: Histograms of resampled importance values for each motif, converted to their standard score, with parameters $n_r = 1000$, $|C_\kappa| = 100$, and $\beta = 5$. Vertical lines correspond to normalized importance values for the cluster of patients that underwent cardiovascular surgery.

1.6. Generating intelligent icons

Let $z_{C_\kappa,\cdot}$ be the vector of normalized importance values within cluster C_κ for all motifs. As noted earlier, because our analysis was limited to length two motifs, the total number of motifs is β^2. Therefore $z_{C_\kappa,\cdot}$ can be reshaped to a square icon with side lengths β. The function named reshape() - implemented in either Matlab or Numpy - was used with $z_{C_\kappa,\cdot}$ and the number of columns and rows as parameters to generate icon matrices. Reshape() in Matlab orders differently than in Numpy, but specifying to use Fortran order in Numpy recreates Matlab behavior. However, since our analysis was done in Numpy, for the sake of convenience Numpy's default behavior was used, which lays motifs out along the icon as shown in figure 8. Pixel colors correspond to $z_{C_\kappa,m}$ values and are capped at $[-3, 3]$, corresponding to -3 and $+3$ standard deviations from the mean.

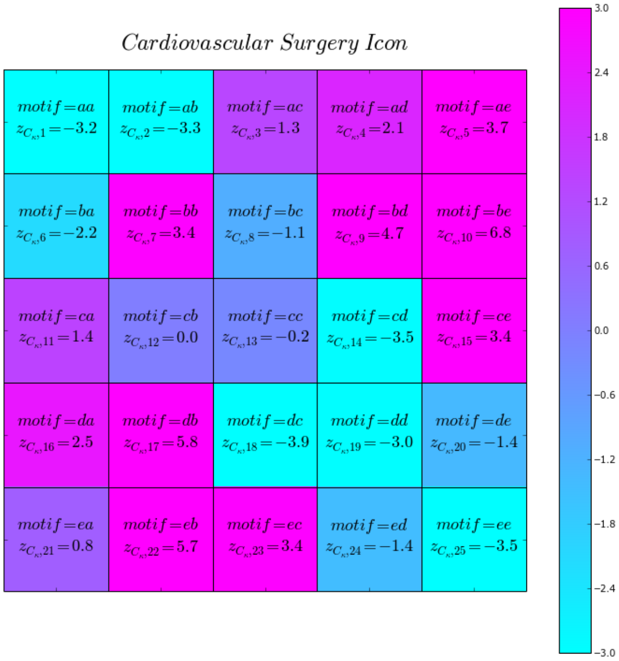


Figure 8: Sample icon for the cluster of patients that underwent cardiovascular surgery, with text overlaying each pixel describing how the color was calculated. **Needed: redo this once decision is made on normalized/unnormalized SAX**

1.7. Sorting icons by similarity

For larger numbers of clusters, it is often useful to determine which clusters are most similar to each other. One option of doing so is to use cosine similarity. For any two clusters C_{κ_1} and C_{κ_2},

$$
\begin{aligned}
similarity(C_{\kappa_1}, C_{\kappa_2}) &= cos\left(\theta_{C_{\kappa_1}, C_{\kappa_2}}\right) \\
&= \frac{z_{C_1,\cdot} \cdot z_{C_2,\cdot}}{\|z_{C_1,\cdot}\| \|z_{C_2,\cdot}\|} \\
&= \frac{\sum_m z_{C_1,m} \times z_{C_2,m}}{\sum_m \left(z_{C_1,m}\right)^2 \times \left(z_{C_2,m}\right)^2}
\end{aligned} \tag{6}
$$

Similarity is thus measured along the interval $[1, -1]$, with 1 being an exact match and -1 being exactly opposite. When plotting icons, a cluster is first chosen and the associated icon plotted. Icons for every other cluster are then plotted in descending similarity from left to right, flowing over to subsequent rows.

2. Examples Needed: redo these once decision is made on normalized/unnormalized SAX. Also want more examples

2.1. Surgical procedure

In this analysis, clusters were defined based on surgical procedure category. Icons are shown in Figure 9 corresponding to $\beta = 2$, Figure 10 to $\beta = 5$, and Figure 11 to $\beta = 10$.

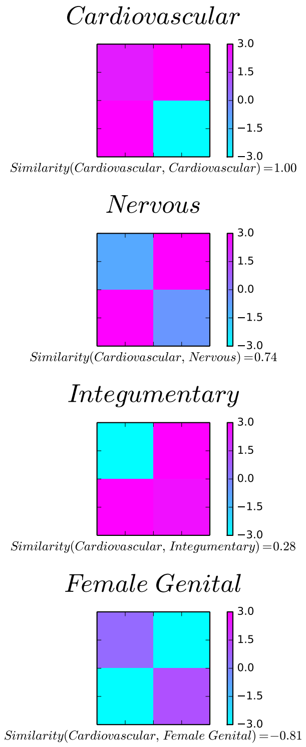
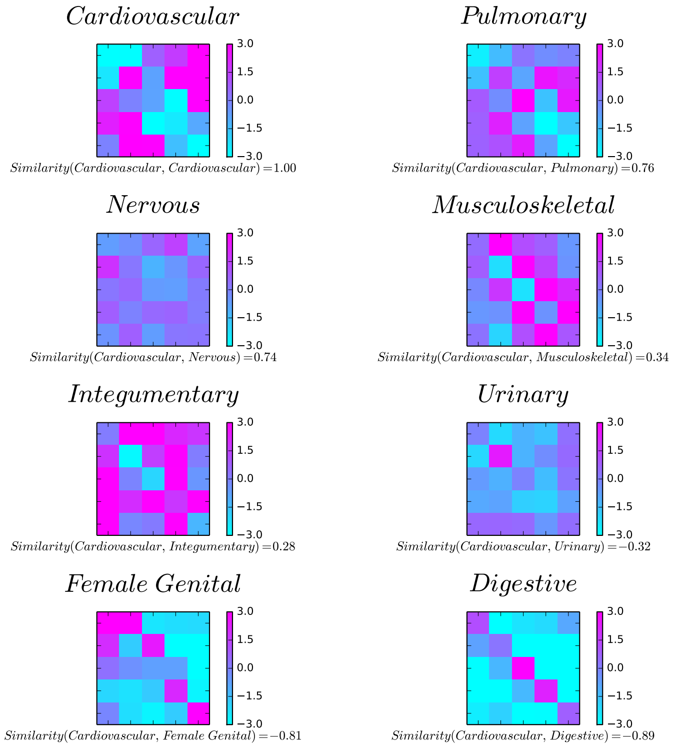


Figure 9: Icons for clusters of patients based on type of surgical procedure, $\beta = 2$. Icons are sorted in descending similarity from the cluster of patients who underwent cardiovascular surgery.



Cardiovascular

Similarity(*Cardiovascular*, *Cardiovascular*) = 1.00

Pulmonary

Similarity(*Cardiovascular*, *Pulmonary*) = 0.76

Nervous

Similarity(*Cardiovascular*, *Nervous*) = 0.74

Musculoskeletal

Similarity(*Cardiovascular*, *Musculoskeletal*) = 0.34

Integumentary

Similarity(*Cardiovascular*, *Integumentary*) = 0.28

Urinary

Similarity(*Cardiovascular*, *Urinary*) = −0.32

Female Genital

Similarity(*Cardiovascular*, *Female Genital*) = −0.81

Digestive

Similarity(*Cardiovascular*, *Digestive*) = −0.89

Figure 10: Icons for clusters of patients based on type of surgical procedure, $\beta = 5$. Icons are sorted in descending similarity from the cluster of patients who underwent cardiovascular surgery.

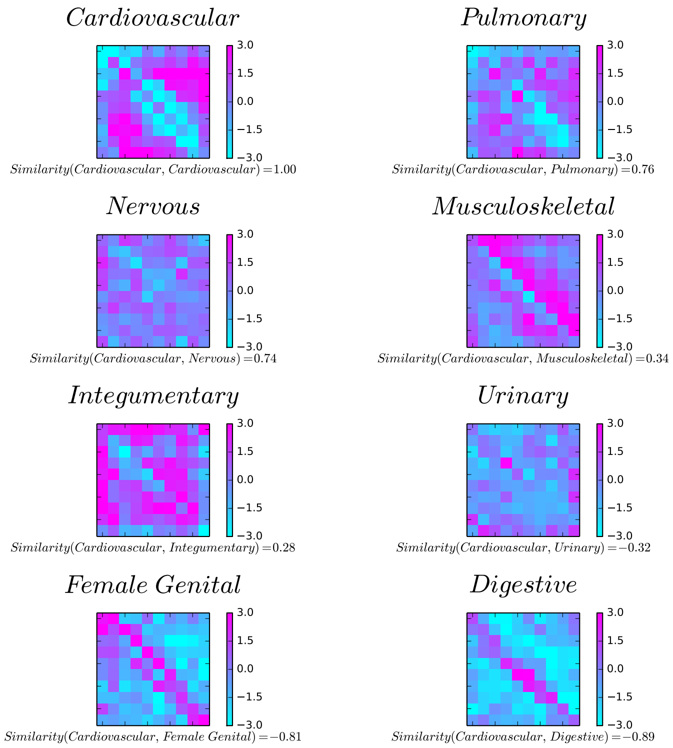


Figure 11: Icons for clusters of patients based on type of surgical procedure, $\beta = 10$. Icons are sorted in descending similarity from the cluster of patients who underwent cardiovascular surgery.