SUPPLEMENTARY MATERIALS: Revealing HIV viral load patterns using unsupervised machine learning and cluster summarization

Samir A. Farooq[1,2], Samuel J. Weisenthal[1,4], Melissa Trayhan[1,2,4], Robert J. White[1,2,4], Kristen Bush[1,4], Peter R. Mariuz[3], and Martin S. Zand[1,2,4,*]

[1]Rochester Center for Health Informatics, 265 Crittenden Boulevard, Rochester, NY 14642-0708, USA
[2]Department of Medicine, Division of Nephrology, 601 Elmwood Ave - Box 675, Rochester, NY 14642, USA
[3]Department of Medicine, Division of Infectious Diseases, University of Rochester Medical Center/Strong Memorial Hospital AIDS Center, 601 Elmwood Ave, Rochester, NY 14642, USA
[4]Clinical and Translational Science Institute, University of Rochester Medical Center, 265 Crittenden Boulevard, Rochester, NY 14642-0708, USA
[*]martin_zand@urmc.rochester.edu

Supplementary Materials

Review of Existing Viral Load Categorization Methods

Greub et. al. LLVR

Greub et. al. were particularly focused on detecting low level viral rebound (LLVR) in patients[1]. The following procedure was used to categorize the patients of their study:
If the patient has two consecutive viral load measurements (VLM) less than 50, within a 24 week period, and they have two VLM after this consecutive pair occurs, then the viral load data for that patient is considered for further analysis. For the patients that meet this criteria, their viral load measurements following the consecutive pair are viewed: if their maximum VLM is greater than 500, then they are categorized as 'Viral Failure', if they are between 51-500 then they are categorized as 'LLVR', otherwise they are labeled as 'DSVL'. Greub et. al. left out the patients which did not meet the consecutive pair criteria from their categorization, hence in our comparative study we will group them under 'Unspecified'.

Rose et. al. SMVL/RMVL

The focus of Rose et. al. was to investigate the use of several frameworks in categorizing suppressed versus not-suppressed viral load[2]. First they omitted the patients from their study whom were virally suppressed at baseline, where they define viral suppression as < 200 copies/mL because they were found to have no substantial variation in their viral loads. In our comparative analysis we label them as 'Baseline < 200'. Then, from the remaining patients they categorize them as either achieving suppression or not-suppressed using an 8 month window centered around month 24 after start of VLM (18-30 months). They considered five different frameworks, which we describe below:

SMVL omit-participant: If the closest VLM to month 24 is < 200 then the patient is labeled as 'Suppressed', otherwise 'Not Suppressed'. However if this closest VLM is outside the range of 18-30 months, then they are labeled as 'Ommitted'.

SMVL set-to-failure: Similar to omit-participant, however if the the closest VLM is outside the range of 18-30 months, then they are labeled as 'Not Suppressed'.

SMVL closest-VL: The patient is labeled according to their closest VLM to month 24 regardless of whether the VLM is contained in the window.

RMVL repeat binary: We do not use this method in our comparative analysis because its purpose is to classify each individual VLM as suppressed or not suppressed (rather than the patient), which is not the goal of this paper.

RMVL repeat continuous: The $\log_{10}$ of the VLMs are modeled as a continuous linear function with its intercept fixed at the baseline viral load. In our implementation we add 1 to each VLM to avoid $\log_{10}(0)$. Then the patient is categorized as 'Suppressed' if the model predicts the patient to have a viral load < 200 copies/mL at month 24, otherwise they are labeled as 'Not Suppressed'.

Terzian et. al. SHVL

The objective of Terzian et. al. was to develop a method of categorizing a patient as DSVL or SHVL for the purpose of monitoring successful ART uptake[3]. Their procedure for categorizing patients is as follows:
If the maximum viral load of the patient is $\leq$ 400 copies/mL then the patient is labeled as 'DSVL'. If the patient instead has two consecutive viral load measurements $\geq$100,000 copies/mL, then the patient is labeled as 'SHVL'. For our analysis all other patients are labeled as 'Unspecified'.

Phillips et. al. Viral Rebound

The aim of Phillips et. al. was to characterize virological response to ART[4]. While the statistical methods proposed by Phillips et. al. went beyond categorizing patients, they composed a method to identify two populations of HIV patients (Viral Failure and Viral Rebound):
Only patients who have at least one VLM within the range 24-40 weeks are included in the categorization, all other patients are labeled as 'Omitted' (similar to SMVL omit-participant). Phillips et. al. chose the 24-40 week range for a different part of their statistical analysis; however, in our categorical implementation, we modify the range to 24-32 weeks to build coherent categories according to the procedure they outline (as we are about to describe). Phillips et. al. choose to use 32 weeks as the point of observing viral load levels because they argue viral load is expected to decline to 500 copies/mL by week 32 if it is going to do so[4]. Thus patients who never achieve VL $<$ 500 copies/mL within 32 weeks are labeled as 'Viral Failure'. If the patient does achieve VL $<$ 500 at one point within 32 weeks, but VLM closest to 32 weeks is $>=$ 500 copies/mL, then this patient is omitted from the categorization. If, however, the patient's closest VLM to 32 weeks is $<$ 500 copies/mL, then if the patient has two consecutive VLM $>=$ 500 copies/mL within 32 weeks, then the patient is labeled as 'Viral Rebound'. Phillips et. al. do not describe them further the patient who did not meet this consecutive pair criteria, however for our implementation we will label them as 'Suppressed'.

Considered but Removed Features

There were several features which were thought to have significance in segregating viral load patterns but did not make it into our feature vector. We had to be careful of extracting features which may be collinear as it would cause a shift in the weighting of features. These collinear features are too many to list here. We mention several excluded features which were generally unreliable or created overlap between classes that should be unrelated:

1. *Minimum.*
2. *Maximum.*
3. *Baseline VL.*
4. *Last VL.*
5. *Rate of Change.* We tried several ways to calculate this feature: 1) Mean of the first derivate of $\overrightarrow{VL}_p$ with respect to $\vec{t}_p$, 2) Median of the first derivative, 3) Rate of change between first and last measurements, 4) Fitting a piecewise regression with one knot and averaging the results of the slope- with the idea that HVLS will have a clear elbow.
6. *Correlation Coefficient.* From the fitted piecewise regression (one knot) we also tried averaging the results of the correlation coefficient, again with the idea that HVLS will have a clear elbow.
7. *Change in Concavity.* Calculated as sum of the absolute changes in concavity with the assumption of equally spaced time (to adjust for issues found in calculating rate of change):

$$\overrightarrow{dVL_i} = \overrightarrow{VL_{i+1}} - \overrightarrow{VL_i}$$

$$\overrightarrow{concavity_i} = \overrightarrow{dVL_{i+1}} - \overrightarrow{dVL_i}$$

$$Change = ||\overrightarrow{concavity}||_1$$

8. *Positive Difference.* We calculate this as the sum of all $\overrightarrow{dVL_i} \geq 0$ normalized by the number of elements satisfying the condition.
9. *Negative Difference.* We calculated this as the sum of all $\overrightarrow{dVL_i} < 0$ normalized by the number of elements satisfying the condition.

Centroid Detection Methodologies

Centroid detection is a problem which several machine learning algorithms attempt to solve, such as Support Vector Machine (SVM), Bayesian Point Machines (BPM), Analytic Centre Machines, k-Means, BIRCH, among others[5,6]. The center of a cloud of samples is generally considered the average[7], but is still a matter of interpretation. We attempt to find cluster centers in seven different ways (Fig S5):

1. *Mean.* Calculated as the average of each dimension (feature).
2. *Median.* Calculated as the median of each dimension (feature).
3. *Best Representative.* The best representative is taken to be the sample whose maximum distance to all the other samples is a minimum.
4. *Bounding Box.* If the cloud of points naturally form an n-dimensional box, then calculating the middle of the minimum and maximum of each dimension would lend itself to be the center representation of the cloud.
5. *Smallest Disk.* If the cloud of points instead naturally form an n-dimensional sphere, then it makes most sense to solve the smallest enclosing disk problem. There are many algorithms that have been proposed to solve this problem[8–10], where we found the best solution in our implementation of the algorithms to be that of Fischer's fast smallest-enclosing ball, where he utilizes the properties of the hull and affine spaces to walk the center to its optimum location[9]. They proposed to initiate the center by choosing a random sample, however this caused the algorithm to yield differing solutions on each run. Through experimentation we found that when we initialized the algorithm using the sample found from best representative method, the algorithm consistently converged

at the solution with the smallest radii, compared to using any other sample as the initial center. Hence in our formulation of Fischer's algorithm we deploy the best representative method as a subroutine for initiation.

6. *Polyhedral.* The trouble with the bounding box and smallest disk approximations for the center is that they are designed to work well under circumstances where the clusters naturally form either a n-dimensional box or sphere respectively. We can relax this constraint by removing this assumption. Rather we propose to find the convex hull of the data points and then find the centroid of the hull, in other words, we find a n-dimensional polyhedron to fit the cloud. Then, given the intersection of the finite half-spaces of the convex hull, we can calculate the exact centroid of the polyhedron- where we use a slight modification of Maire's algorithm[5]. The hyperplane equations are calculated in Python using the `ConvexHull` and `Delaunay` packages within `SciPy`.

7. *Push and Pull.* The six methods mentioned are not designed to maximize the distance between each cluster center while minimizing the distance within the cluster. In other words, the above six methods calculate the center of each point cloud without any dependencies on the other clouds. We propose a push and pull method, inspired by force-directed graph drawing using Fruchterman-Reingold's algorithm[11] to include dependencies on the center of each cloud to improve performance on centroid learning. That is, let every sample within the cluster have a pulling force on the center following Hooke's Law, $F = kd$, and let every sample outside the cluster have a pushing force on the center following a modified version of Coulomb's Law, $F = k_e \frac{q_1 q_2}{d^2}$. In our case, d represents distance between the center and sample, and the constants k, k_e, q_1, and q_2, equal to 1 in order to yield equally weighted pulling and pushing. For each cluster, we initialize the center as the mean of the cluster followed by iteratively finding the center until achieving a dynamic equilibrium. We initiate at the mean of the cluster samples because the spring-like pulling of each sample on the center is the same as the mean of the samples pulling on the center with a spring constant equal to the number of samples (proved below).

Proposition *The combined spring-like pull of each sample, S_i, on a center, C, with a spring constant of 1, is same as a spring-like pull of the mean of the samples, $\overline{S}$, on C with a spring constant equal to the number of samples pulling on C.*

Proof. Let the sample S_i and the center C be n-dimensional vectors such that S_i has a spring-like pulling force on C with a spring constant of 1, that is, $F_i = \frac{||S_i - C||}{||S_i - C||}(S_i - C) = S_i - C$, where $\frac{1}{||S_i - C||}(S_i - C)$ is the direction of the force and $||S_i - C||$ is the magnitude, or distance between S_i and C. Notice the mean of the samples $\overline{S} = \frac{\sum_{i=1}^{n} S_i}{n}$, where n is the total number of samples, then observe that the sum of the spring-like forces acting on the center is given by

$$\begin{aligned} F &= \sum_{i=1}^{n} F_i = \sum_{i=1}^{n} (S_i - C) \\ &= \sum_{i=1}^{n} S_i - \sum_{i=1}^{n} C \\ &= (\sum_{i=1}^{n} S_i) - nC \\ &= n(\frac{\sum_{i=1}^{n} S_i}{n} - C) \\ &= n(\overline{S} - C) \end{aligned}$$

This yields we have that the sum of all the spring-like forces on C is the same as the mean average of the samples having a spring-like pull on C with a spring constant of n, which is the number of samples pulling on C. □

Observe that some clusters have a low number of samples pulling at the center relative to all the samples having a pushing effect. Hence, if n is small, then we will find that the center is pushed more than desired, and if n is large, then the pull towards the average will be too strong and hence the center will be no different from the average. We propose to replace n with the number of samples having a pushing effect on the center.

References

[1] Greub G, Cozzi-Lepri A, Ledergerber B, Staszewski S, Perrin L, Miller V, Francioli P, Furrer H, Battegay M, Vernazza P, Bernasconi E, Günthard HF, Hirschel B, Phillips AN, Telenti A. AIDS. 2002 sep;16(14):1967–1969. Available from: https://doi.org/10.1097%2F00002030-200209270-00017.

[2] Rose CE, Gardner L, Craw J, Girde S, Wawrzyniak AJ, Drainoni ML, Davila J, DeHovitz J, Keruly JC, Westfall AO, Marks G. PLOS ONE. 2015 jun;10(6):e0130090. Available from: https://doi.org/10.1371%2Fjournal.pone.0130090.

[3] Terzian AS, Bodach SD, Wiewel EW, Sepkowitz K, Bernard MA, Braunstein SL, Shepard CW. PLoS ONE. 2012 jan;7(1):e29679. Available from: https://doi.org/10.1371%2Fjournal.pone.0029679.

[4] Andrew N Phillips RW Schlomo Staszewski. JAMA. 2001 nov;286(20):2560. Available from: https://doi.org/10.1001%2Fjama.286.20.2560.

[5] Maire F. An algorithm for the exact computation of the centroid of higher dimensional polyhedra and its application to kernel machines. In: Third IEEE International Conference on Data Mining. IEEE Comput. Soc; 2003. Available from: https://doi.org/10.1109%2Ficdm.2003.1250988.

[6] Han J, Pei J, Kamber M. Data mining: concepts and techniques. Elsevier; 2011.

[7] Abdi H. Wiley Interdisciplinary Reviews: Computational Statistics. 2009 sep;1(2):259–260. Available from: https://doi.org/10.1002%2Fwics.31.

[8] Welzl E. Smallest enclosing disks (balls and ellipsoids). In: New Results and New Trends in Computer Science. Springer-Verlag; 1991. p. 359–370. Available from: https://doi.org/10.1007%2Fbfb0038202.

[9] Fischer K, Gärtner B, Kutz M. Fast Smallest-Enclosing-Ball Computation in High Dimensions. In: Algorithms - ESA 2003. Springer Berlin Heidelberg; 2003. p. 630–641. Available from: https://doi.org/10.1007%2F978-3-540-39658-1_57.

[10] Nielsen F, Nock R. Approximating Smallest Enclosing Balls. In: Computational Science and Its Applications – ICCSA 2004. Springer Berlin Heidelberg; 2004. p. 147–157. Available from: https://doi.org/10.1007%2F978-3-540-24767-8_16.

[11] Kobourov SG. arXiv preprint arXiv:12013011. 2012;.