Supplementary Information

Musical Gene Expression Sonification Method Summary

Biological time series expression data corresponding to an individual's genetic or proteomic material is submitted to principal component analysis, linked by protein interaction connections. The resulting principal components are each mapped to a musical frequency, duration, and volume to produce a harmonious chord for each time point and a musical phrase or combination of phrases. The mapping of frequencies is based on the Pythagorean tuning laws of harmony versus dissonance, constraining simultaneous frequencies (i.e. chords) to intervals that are necessarily harmonious with each other, such as thirds, fifths, and octaves. Duration of each chord is determined by the logarithm of the time delay from one experiment of the biological time series to the next.

Musical Gene Expression Sonification Method Details

In order for harmoniousness to be used meaningfully as a discriminatory tool, not only must the comparative sonification always be harmonious when the control is compared to normal data, but also it must be inharmonious when the control is compared to abnormal (e.g. disease) data. The challenge is in solving the fundamental problem, relevant to most scientific sonification, of finding suitable mappings between the space of data and the space of sounds [1].

In our work, we have implemented the sonification process by applying principal component analysis (PCA) to gene expression data, projecting the data via the eigenvectors produced, and calculating audio frequencies based on the resulting linear combinations between the PCA eigenvectors and the expression data. The number of principal components was selected using the scree plot method [2]. The PCA-adjusted expression values are mapped to frequencies as described below. Every note thereafter is determined relative to the base note frequency set at t_0. Specifically, for each experiment at time t_N, we take the logarithm of the ratio between the PCA- adjusted expression value $PCAx_{N,}$ and the initial value $PCAx_{0,}$ at time t_0: $V = \log(PCAx_{N,}/PCAx_{0,})$.

This normalization process reveals how the PCA-adjusted values change across time, relative to the initial value. We use this normalized value V to calculate the new frequency. First, we determine how many octaves to move up (or down, if the value is negative). The number of octaves is calculated to be the minimum possible N such that: $1 \leq V*8^N$. The value V_1 given by $V*8^N$ determines whether to move up another third or fifth beyond the octaves (eight notes) calculated. If V_1 modulus $8 \geq 5$, which we denote as case A, then the frequency is moved by another fifth. Otherwise if V_1 modulus $8 \geq 3$, denoted as case B, then the frequency is instead moved by another third. If V_1 modulus $8 < 3$, then the frequency is determined simply by the number of octaves calculated. In limiting the possible intervals to only octaves, fifths, and thirds, we effectively normalize the control data to intervals that are always harmonious with each other.

We determine the frequencies for the experimental data set by comparing the PCA-adjusted expression values to their analogs (same gene and time point) in the control data set, but the calculation of the number of octaves, fifths, and thirds is otherwise the same as for the control data. Thus, the experimental data map to the same notes as the control data only if it fluctuates in the same way.

We have normalized the control notes to harmonious intervals. Yet, the experimental test conditions are free from this basis. So, if the experimental data moves differently from the control data, the experimental notes may become inharmonious. For the experimental data set, we calculate the number of musical seconds to add to the resulting frequency as V_1 modulus 8 – 5 for case A and $V1$ modulus 8 – 3 for case B.

To translate the number of octaves, fifths, thirds, and seconds into a musical frequency, we apply the mathematics of Pythagorean tuning. Given a prior frequency F_0, we multiply F_0 by 2 to obtain a new frequency F_1 that is an octave above F_0. Multiplying F_0*1.5 produces F_1 that is a fifth above F_0, and F_0*1.2 produces F_1 that is a third above F_0. The result is music that sounds harmonious across time for normal data and inharmonious as the dynamics change due to perturbation/disease.

References

1. Kaper H.G. and Tipei S. Formalizing the Concept of Sound. *Proc. Int'l Computer Music Conference '99*, Beijing (October 1999), pp. 387-390.

2. Jolliffe I.T. *Principal component analysis*, xxix, 487 p. (Springer, New York, 2002).

Supplementary Figures

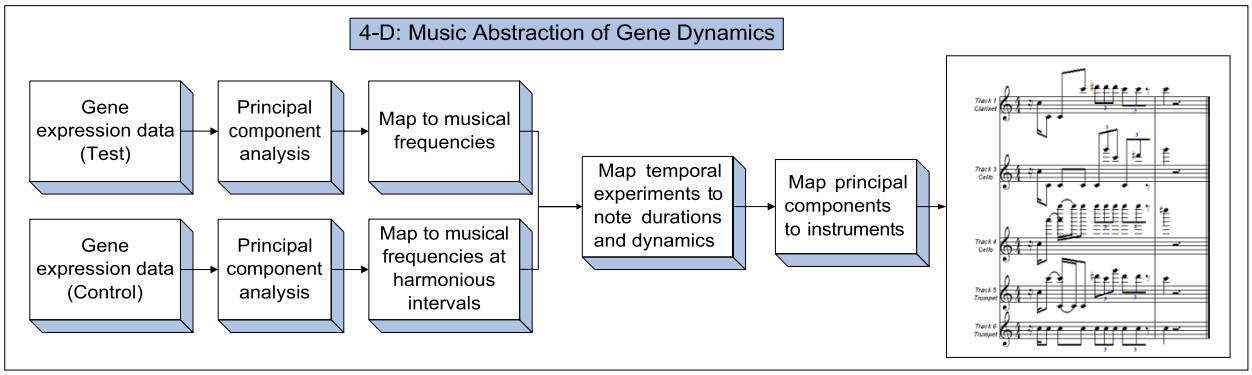

Figure S1: Method overview.

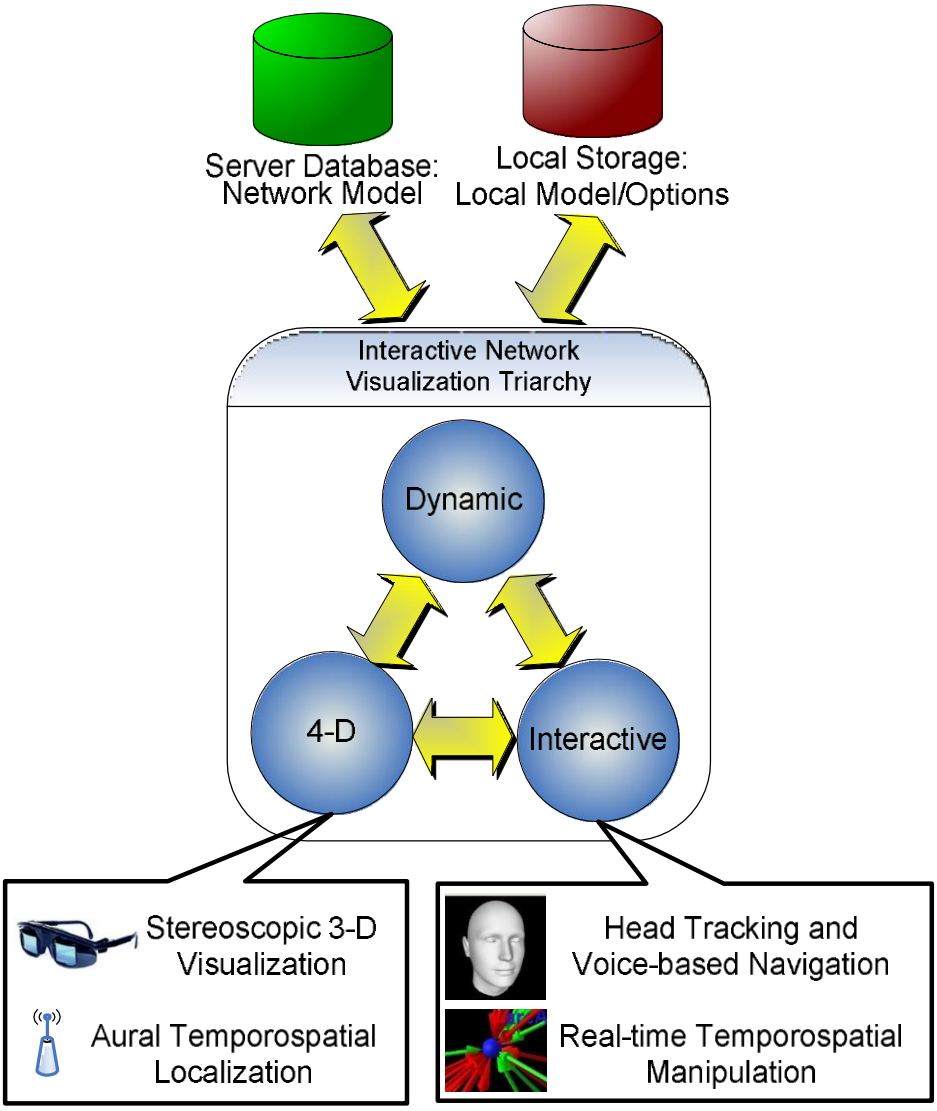

Figure S2: Architecture overview