1 Other ML Methods

1.1 K-nearest neighbors (KNN)

1.1.1 Approach

Given a training set of periodic and aperiodic signals, K-nearest neighbors (KNN) can be performed to determine the class of a signal. The output is the percentage of its K closest neighbors that are periodic.

1.1.2 Calculating p-values

Computing p-values for periodic/aperiodic classification is similar to the case of DNNs.

1.1.3 Period Estimation

A slightly different version of KNN is used to determine the period for signals that are classified as periodic. In this case, the training set consists of periodic signals only and the estimated period is the average of the periods of the signal's K closest neighbors.

1.1.4 Hyperparameter Selection

To determine how large the training set needs to be and the value of K (the number of nearest neighbors being used), a grid search is done for KNN trained on the $\text{BioCycle}_{\text{Func}}$ dataset and tested on the $\text{BioCycle}_{\text{Gauss}}$ (48_8) dataset. The reason is that this corresponds to one of the harder scenarios because the testing set is different from the training set and the dimensionality of the data is on the larger side. The results are shown in Tables 1 and 2. From the results we see that $K = 100$ for 1,000,000 provides the best results and these are comparable to those of the DNN.

Table 1. AUC on the $\text{BioCycle}_{\text{Gauss}}$ (48_4) for the KNN_{G} method.

		Training Data			
		1000	10000	100000	1000000
K	5	0.92	0.94	0.94	0.94
	10	0.93	0.95	0.95	0.95
	100	0.91	0.95	**0.96**	**0.96**

Table 2. R^2 for the period on the $\text{BioCycle}_{\text{Gauss}}$ (48_4) for the KNN_{G} method.

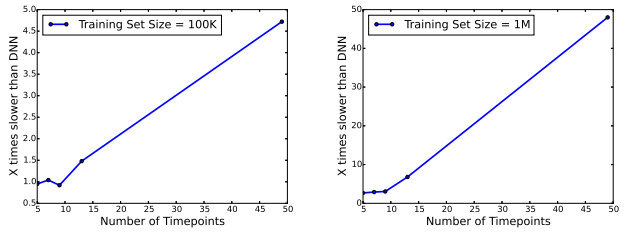
		Training Data			
		1000	10000	100000	1000000
K	5	0.69	0.74	0.75	0.76
	10	0.72	0.77	0.78	0.78
	100	0.71	0.77	0.79	**0.80**

1.1.5 Memory and Time Consumption

All the information that the deep neural network (DNN) has is contained in its training dataset. In Section 1.3, we see that KNN performs about the same as DNN when using a training set of 1,000,000 examples. However, there are both memory and run time issues that make deep learning more efficient.

Figure 1 shows an analysis of how long it takes for KNN to run on the $\text{BioCycle}_{\text{Func}}$ test set of 20,000 examples for two values of the size of the training set. With a training set of size 1,000,000, using KNN is 48 times slower than using DNN. The actual runtime in this case is 4 hours for the KNN method and 5 minutes for the DNN method.

The size of the DNN model consists of its parameters (weights and biases), however, the size of the KNN model consists of the entire training set. So, for example, if we use the $\text{BioCycle}_{\text{Gauss}}$ (48_4) dataset, the number of parameters in the DNN (3 hidden layers, 100 neurons) is about



(a) With a training set of size 100,000. (b) With a training set size of 1,000,000.

Fig. 1: The performance of the KNN compared to the performance of the DNN. The y axis is the time it takes to run the KNN method over the time it takes to run the DNN method.

21,700 floating point numbers. However, for the KNN model, each time series in the training set has 13 floating point numbers. With a training set of size 1,000,000 this corresponds to a size of 13,000,000 floating point numbers. Therefore, the KNN model takes 600 times more memory space.

1.2 Gaussian Processes

1.2.1 Approach

We have a model for periodic signals M_p and a model for aperiodic signals M_a. Then, given a signal $\mathbf{s}$, we want to compare the probabilities $p(M_p|\mathbf{s})$ and $p(M_a|\mathbf{s})$ to determine if the signal is periodic or aperiodic.

We need both a mean and a covariance matrix. We assume that the data has a mean of zero. The probability density function given a signal $\mathbf{s}$ is defined by equation 1, where Σ is the covariance matrix and k is the dimension of the data (i.e. how many timepoints there are in the signal).

$$p(\mathbf{s}|\Sigma) = (2\pi)^{-\frac{k}{2}}|\Sigma|^{-\frac{1}{2}}exp(-\frac{1}{2}\mathbf{s}^T\Sigma^{-1}\mathbf{s}) \tag{1}$$

Kernel functions are used to build these covariance matrices for the periodic and aperiodic signals. When using a kernel function, the entries at timepoints x and x' are determined by a kernel function $k(x, x')$. We use equation 2 for the periodic covariance matrix Σ_p and equation 3 for the aperiodic covariance matrix Σ_a.

$$k_p(x, x') = exp(\frac{-sin^2(|\pi\frac{1}{p}(x - x')|)}{2l_p^2}) + \sigma_p^2\delta(x, x') \tag{2}$$

$$k_a(x, x') = exp(\frac{-(x - x')^2}{2l_a^2}) + \sigma_a^2\delta(x, x') \tag{3}$$

The l parameter controls how strong the covariance is between two different points and the parameter σ^2 controls how noisy the data is believed to be. The parameter p in equation 2 is the period of the signal being modeled by $k_p(x, x')$. The period parameter p is set prior to optimizing the other parameters and is determined by the range of periods being searched for. The periodic model M_p is a mixture of models where the covariance matrix for each model is obtained with the kernel function in equation 2 for different values of p. The values of p start from the beginning of the range of periods being searched for and is increased by 1 until p reaches the end of the range.

1.2.2 Hyperparameter Selection

The parameters l_p, σ_p, l_a, and σ_a are determine by a random grid search. Every new proposed set of parameters is used to give p-values for the training dataset. The parameters that give the highest AUC are used. A total of 100 different sets of parameters are tried.

1.2.3 Calculating p-values

The approach to getting p-values for this model is similar to the approach used in the case of the DNN. The only difference is that the output is now $\frac{p(M_p|\mathbf{s})}{p(M_a|\mathbf{s})+p(M_p|\mathbf{s})}$.

1.2.4 Period Estimation

To determine what the period is, we first find the model in the mixture of models for M_p that has the highest density given the input. The corresponding period associated with that model is the estimated period.

1.3 Comparisons

We compare all these different machine learning methods on the BioCycle dataset. The comparisons are carried both when the methods are trained on the $\text{BioCycle}_{\text{Func}}$ dataset and when they are trained on the $\text{BioCycle}_{\text{Gauss}}$ dataset. In the main case, the periodic data has periods between 20 and 28, like the experiments in the main text. The results are shown in Tables 3, 4, 5, and 6. The results show in this case that the DNN and KNN approaches achieve the best performance and outperform the Gaussian process approach.

Table 3. AUC performance on synthetic data.

	GP_F	GP_G	KNN_F	KNN_G	DNN_F	DNN_G
BC_F (24_4)	0.90	0.90	**0.92**	0.91	**0.92**	0.91
BC_F(24_6)	0.84	0.84	**0.85**	0.84	**0.85**	0.84
BC_F (48_4)	0.96	0.96	**0.97**	0.96	**0.97**	0.96
BC_F (48_8)	0.88	0.88	**0.89**	**0.89**	**0.89**	**0.89**
BC_F (24_U)	0.88	0.88	**0.89**	0.88	**0.89**	0.88
BC_F (48_U)	0.92	0.92	**0.94**	0.93	**0.94**	0.93
BC_G (24_4)	0.91	0.92	0.92	**0.94**	0.92	**0.94**
BC_G (24_6)	0.87	0.87	0.87	0.88	0.88	**0.89**
BC_G (48_4)	0.96	0.96	0.96	**0.97**	**0.97**	**0.97**
BC_G (48_8)	0.91	0.91	0.92	**0.93**	**0.93**	**0.93**
BC_G (24_U)	0.91	0.91	0.91	**0.92**	0.91	**0.92**
BC_G (48_U)	0.94	0.94	0.95	**0.96**	0.95	**0.96**

Table 4. Coefficients of determinations (R^2) for the periods.

	GP_F	GP_G	KNN_F	KNN_G	DNN_F	DNN_G
BC_F (24_4)	0.24	0.21	**0.31**	0.27	**0.31**	0.27
BC_F(24_6)	0.18	0.18	**0.22**	0.19	**0.22**	0.19
BC_F (48_4)	0.64	0.65	**0.75**	0.74	0.74	0.73
BC_F (48_8)	0.46	0.47	**0.57**	0.56	**0.57**	0.55
BC_F (24_U)	0.21	0.21	**0.28**	0.24	**0.28**	0.24
BC_F (48_U)	0.52	0.53	**0.63**	0.60	0.62	0.60
BC_G (24_4)	0.25	0.26	0.34	0.39	0.35	**0.40**
BC_G (24_6)	0.28	0.28	0.31	**0.36**	0.32	**0.36**
BC_G (48_4)	0.68	0.68	0.80	**0.81**	0.80	**0.81**
BC_G (48_8)	0.51	0.52	0.66	**0.69**	0.67	**0.69**
BC_G (24_U)	0.27	0.26	0.32	0.36	0.32	**0.37**
BC_G (48_U)	0.64	0.63	0.73	**0.75**	0.73	**0.75**

2 Distribution of p-values for BIO_CYCLE

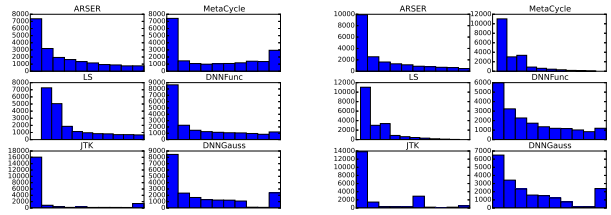
The distribution of the p-values is shown in Figure 2.

Table 5. Coefficients of determinations (R^2) for the lags.

	GP_F	GP_G	KNN_F	KNN_G	DNN_F	DNN_G
BC_F (24_4)	0.46	0.41	**0.49**	0.48	**0.49**	**0.49**
BC_F(24_6)	0.40	0.40	**0.45**	0.44	**0.45**	0.43
BC_F (48_4)	0.50	0.50	**0.53**	0.51	0.52	0.51
BC_F (48_8)	0.40	0.40	**0.42**	0.41	**0.42**	0.41
BC_F (24_U)	0.44	0.44	**0.47**	**0.47**	**0.47**	**0.47**
BC_F (48_U)	0.44	0.44	**0.49**	0.48	**0.49**	0.48

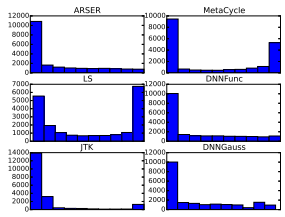
Table 6. Coefficients of determinations (R^2) for the amplitudes.

	GP_F	GP_G	KNN_F	KNN_G	DNN_F	DNN_G
BC_F (24_4)	0.80	0.80	**0.81**	**0.81**	**0.81**	**0.81**
BC_F(24_6)	0.79	0.79	**0.80**	**0.80**	**0.80**	**0.80**
BC_F (48_4)	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**
BC_F (48_8)	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**
BC_F (24_U)	0.79	0.79	**0.80**	**0.80**	**0.80**	**0.80**
BC_F (48_U)	**0.77**	**0.77**	**0.77**	**0.77**	**0.77**	**0.77**

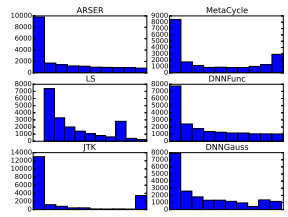


(a) $\text{BioCycle}_{\text{Form}}$ (24_4)

(b) $\text{BioCycle}_{\text{Form}}$ (24_6)



(c) $\text{BioCycle}_{\text{Form}}$ (48_4)



(d) $\text{BioCycle}_{\text{Form}}$ (48_8)

Fig. 2: Histograms of p-values.

3 Detecting Periods of 8 and 12 Hours

$\text{BioCycle}_{\text{Form}}$ and $\text{BioCycle}_{\text{Gauss}}$ datasets can be generated for different period ranges. In the main text, this range is 20-28, focusing on detecting signals with periods of 24 hours. Since there have also been genes discovered with periods of 12 and 8 hours, we generate $\text{BioCycle}_{\text{Form}}$ and $\text{BioCycle}_{\text{Gauss}}$ datasets with period ranges from 10-14 and from 7-9 to focus on the 12 and 8 hour periods, respectively, corresponding to the second and third harmonics. The results for detecting periods of 12 hours are shown in Ttables 7, 8, 9, and 10. The results for detecting periods of 8 hours are shown in tables 11, 12, 13, and 14.

The JTK and ARSER methods did not run on these datasets, so we only compare to LS and MetaCycle. We also note that the meta predictor MetaCycle, which uses JTK, ARSER, and LS, only chose to use LS for these datasets.

The results show that BIO_CYCLE is the best choice in almost all cases.

Table 7. AUC performance on synthetic data. Periodic data has periods between 10 and 14.

	LS	MC	DNN_F	DNN_G
BC_F (24_4)	0.89	0.89	0.87	**0.90**
BC_F(24_6)	0.83	0.83	**0.86**	**0.86**
BC_F (48_4)	0.94	0.94	**0.96**	0.95
BC_F (48_8)	0.81	0.81	**0.85**	0.83
BC_F (24_U)	0.86	0.86	**0.87**	**0.87**
BC_F (48_U)	0.90	0.90	**0.92**	0.91
BC_G (24_4)	0.93	0.93	0.92	**0.94**
BC_G (24_6)	0.88	0.88	**0.91**	**0.91**
BC_G (48_4)	0.95	0.95	0.96	**0.97**
BC_G (48_8)	0.88	0.88	0.89	**0.90**
BC_G (24_U)	0.91	0.91	**0.92**	**0.92**
BC_G (48_U)	0.93	0.93	**0.94**	**0.94**

Table 8. Coefficients of determinations (R^2) for the periods. Periodic data has periods between 10 and 14.

	LS	MC	DNN_F	DNN_G
BC_F (24_4)	0.50	0.50	**0.57**	**0.57**
BC_F(24_6)	0.00	0.00	**0.01**	**0.01**
BC_F (48_4)	0.79	0.79	**0.85**	0.84
BC_F (48_8)	0.60	0.60	**0.70**	0.69
BC_F (24_U)	0.29	0.29	**0.47**	0.45
BC_F (48_U)	0.52	0.52	**0.66**	0.64
BC_G (24_4)	0.57	0.57	0.66	**0.68**
BC_G (24_6)	0.00	0.00	**0.01**	**0.01**
BC_G (48_4)	0.79	0.79	0.86	**0.87**
BC_G (48_8)	0.66	0.66	0.77	**0.78**
BC_G (24_U)	0.43	0.43	0.57	**0.60**
BC_G (48_U)	0.60	0.60	0.72	**0.74**

Table 9. Coefficients of determinations (R^2) for the lags. Periodic data has periods between 10 and 14.

	LS	MC	DNN_F	DNN_G
BC_F (24_4)	0.13	0.24	**0.39**	**0.39**
BC_F(24_6)			0.00	0.00
BC_F (48_4)	0.00	0.00	**0.47**	0.44
BC_F (48_8)	0.03	0.06	**0.33**	0.30
BC_F (24_U)	0.13	0.30	**0.34**	0.32
BC_F (48_U)	0.00	0.00	**0.34**	0.33

Table 10. Coefficients of determinations (R^2) for the amplitudes. Periodic data has periods between 10 and 14.

	LS	MC	DNN_F	DNN_G
BC_F (24_4)	0.61	**0.88**	0.86	0.86
BC_F(24_6)			**0.59**	0.58
BC_F (48_4)	0.42	0.58	0.83	**0.84**
BC_F (48_8)	0.57	0.56	**0.77**	**0.77**
BC_F (24_U)	0.60	0.75	**0.82**	**0.82**
BC_F (48_U)	0.44	0.63	**0.81**	**0.81**

Table 11. AUC performance on synthetic data. Periodic data has periods between 7 and 9.

	LS	MC	DNN_F	DNN_G
BC_F (24_4)	0.90	0.90	**0.92**	**0.92**
BC_F(24_6)	0.61	0.61	**0.80**	0.79
BC_F (48_4)	0.95	0.95	0.95	**0.96**
BC_F (48_8)	0.59	0.59	**0.85**	0.81
BC_F (24_U)	0.85	0.85	**0.88**	0.87
BC_F (48_U)	0.90	0.90	**0.92**	**0.92**
BC_G (24_4)	0.93	0.93	**0.94**	**0.94**
BC_G (24_6)	0.69	0.69	0.81	**0.85**
BC_G (48_4)	0.95	0.95	0.94	**0.97**
BC_G (48_8)	0.66	0.66	0.66	**0.73**
BC_G (24_U)	0.90	0.90	0.91	**0.92**
BC_G (48_U)	0.92	0.92	0.93	**0.94**

Table 12. Coefficients of determinations (R^2) for the periods. Periodic data has periods between 7 and 9.

	LS	MC	DNN_F	DNN_G
BC_F (24_4)	0.01	0.01	**0.03**	0.01
BC_F(24_6)	0.36	0.36	**0.44**	0.42
BC_F (48_4)	0.02	0.02	**0.14**	0.10
BC_F (48_8)	0.00	0.00	**0.02**	0.01
BC_F (24_U)	0.40	0.40	**0.50**	0.48
BC_F (48_U)	0.50	0.50	**0.65**	0.63
BC_G (24_4)	0.00	0.00	0.01	**0.02**
BC_G (24_6)	0.46	0.46	0.54	**0.56**
BC_G (48_4)	0.04	0.04	0.14	**0.16**
BC_G (48_8)	0.01	0.01	0.02	**0.04**
BC_G (24_U)	0.48	0.48	0.59	**0.62**
BC_G (48_U)	0.59	0.59	0.74	**0.76**

Table 13. Coefficients of determinations (R^2) for the lags. Periodic data has periods between 7 and 9.

	LS	MC	DNN_F	DNN_G
BC_F (24_4)	0.00	0.00	0.00	**0.01**
BC_F(24_6)			**0.45**	0.44
BC_F (48_4)	0.00	0.00	0.00	0.00
BC_F (48_8)	0.00	0.00	0.00	0.00
BC_F (24_U)	0.07	0.12	**0.26**	0.25
BC_F (48_U)	0.01	0.02	**0.17**	**0.17**

Table 14. Coefficients of determinations (R^2) for the amplitudes. Periodic data has periods between 7 and 9.

	LS	MC	DNN_F	DNN_G
BC_F (24_4)	0.52	0.49	**0.69**	**0.66**
BC_F(24_6)			**0.80**	**0.80**
BC_F (48_4)	0.42	0.10	0.57	**0.59**
BC_F (48_8)	**0.47**	0.01	0.03	0.02
BC_F (24_U)	0.59	0.61	**0.74**	**0.74**
BC_F (48_U)	0.46	0.62	**0.77**	**0.77**

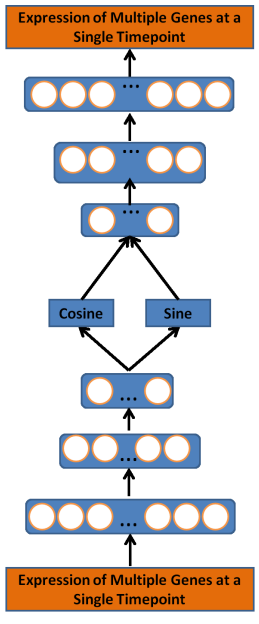


Fig. 3: A visualization of an autoencoder with a cosine and sine unit as the bottleneck.

4 Autoencoders and Manifold Learning

We also investigated an alternative unsupervised manifold learning approach for automatically extracting the time associated with a high-throughput transcriptomic measurement taken at a single timepoint. The basic idea is to use a compressive autoencoder with a bottleneck consisting of two special units (Figure 3). The autoencoder can be applied to the full sets of measurements, or to a subset (e.g. the core clock genes). In trying to reconstruct the input data in the final output layer, the autoencoder must compress the data through these two units optimally in a way that hopefully correspond to the cosine and sine of the phase angle, up to a circular shift. If the activations of these two units are S_1 and S_2, then their two outputs are given by: $S_1/\sqrt{S_1^2+S_2^2}$ and $S_2/\sqrt{S_1^2+S_2^2}$. The autoencoder can be trained using large amounts of unlabeled data, for instance taken in GEO. While this approach generates interesting results, the supervised approach used to train BIO_CLOCK so far yields better results.