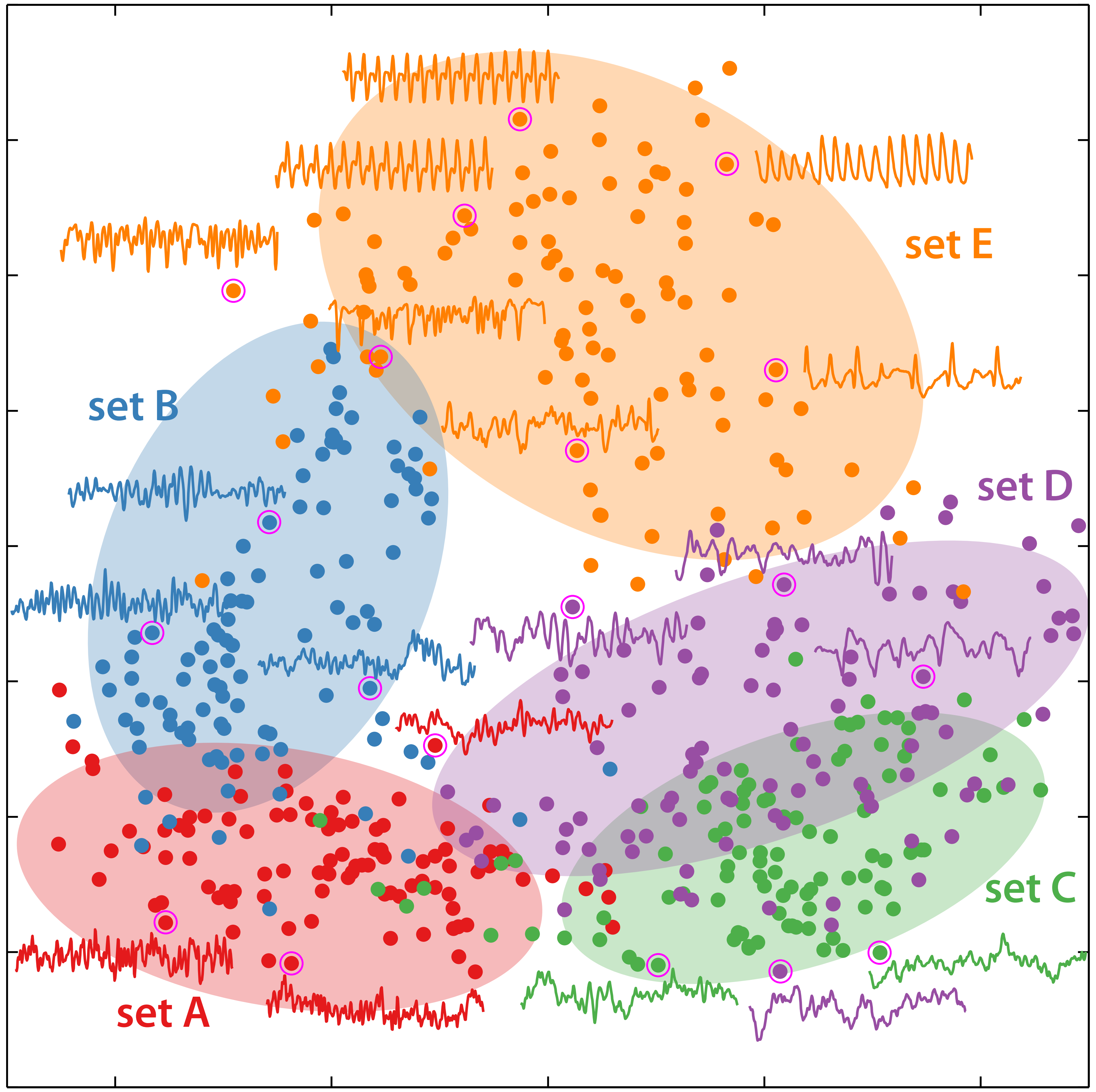
Characterizing neural dynamics using highly comparative time-series analysis



CNS2022 Tutorial, Saturday July 16, 2022

Ben Fulcher, Annie Bryant, Trent Henderson
Dynamics and Neural Systems Lab, School of Physics, The University of Sydney.



Today

- **Intro to the highly comparative approach (30 min)**
    - Surveys of the scientific literature of methods allow us to compare across a diverse literature.
- **Software implementations of this approach**
    - Ben Fulcher (10 min): Features of univariate time series: *hctsa* (and *catch22*) in Matlab.
    - Trent Henderson (10 min): Analyses using open-source feature sets in R with *theft.*
    - Annie Bryant (10 min): Features of pairwise interactions between time series with *pyspi*.
- **-Break-** (30 min, to align with scheduled break time: 10:30-11:00)
- **Demos (45 min)**
    - *hctsa (15 min), theft (15 min), pyspi (15 min).*
- **Interactive session *(45* min*)***
    - Work through sample datasets (or your own data) together.

Disagreements about methods are not uncommon

Part 1: Time series are measured, simulated, studied, and analyzed across a wide variety of disciplines

The structure in our data is often similar

The types of methods we use are often similar

We don’t talk to each other so much

What types of methods have scientists developed?

What types of data do scientists study?

Fulcher et al. (2013). Highly comparative time-series analysis: the empirical structure of time series and their methods. *J. Roy. Soc. Interface*, **10:** 20130048.

How do scientists uncover relationships?

First measure comprehensively, then search for simplifying structure

Wanna learn about butterfly diversity?

Frolic in field with net

before you know it →

Butterfly collection

Wanna learn about time series and their methods?

1– Collect many scientific time series

2– Collect many scientific time-series analysis methods

3– Use properties of data as measure by the methods to **organize our data**

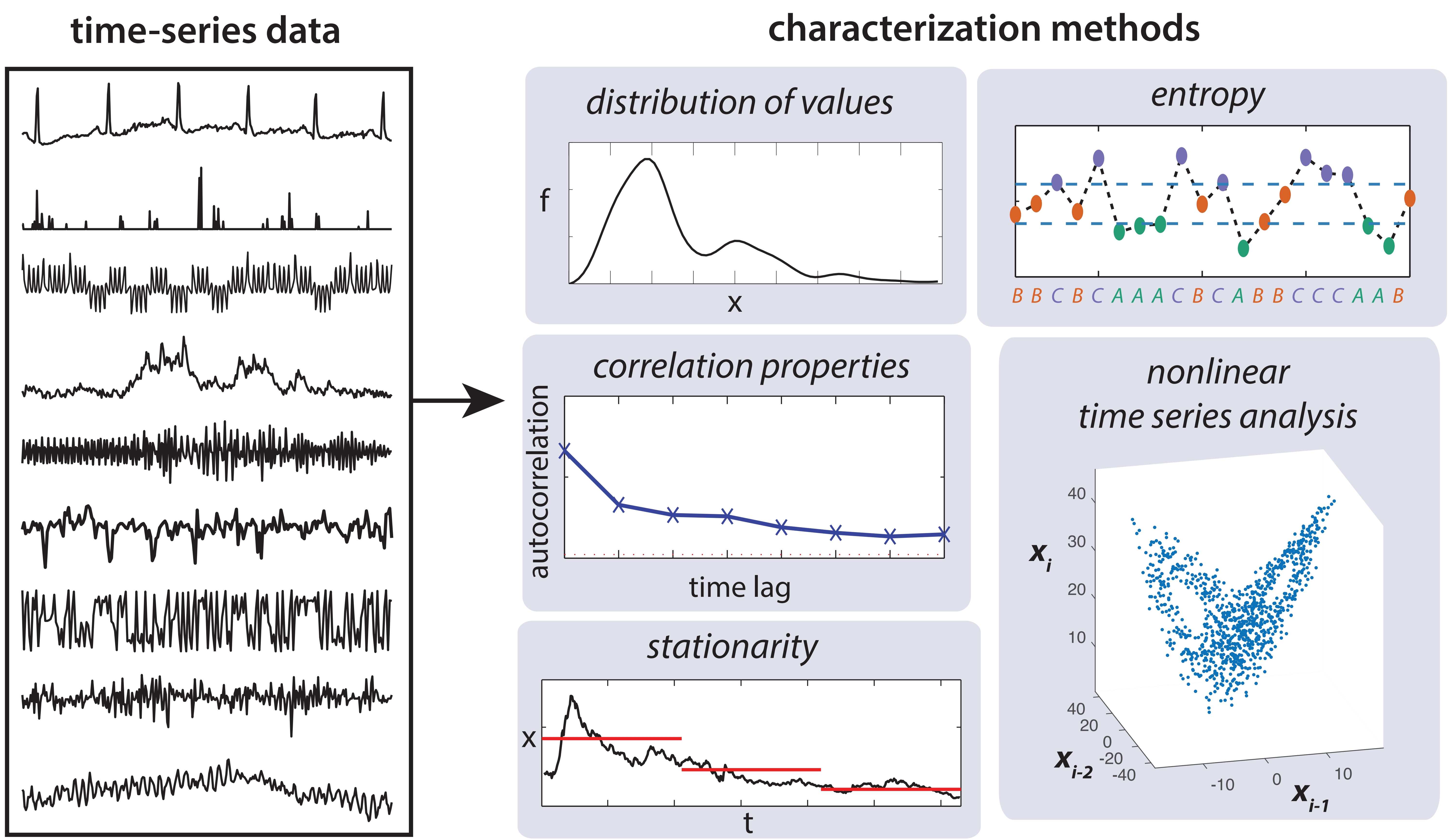
4– Use performance of methods on data to **organize our methods**

Many of our measurements of the world are in the form of **time series**

Repeated measurements of some system over time: $(x_1, x_2, x_3, \ldots)$

Characterizing univariate time series using features

How can I reduce complex time-varying patterns to informative summary statistics? $f : \mathbb{R}^N \to \mathbb{R}$



How fast is it varying?

$$\tilde{x}_k = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} x_n e^{2\pi i k n/N}$$

How variable?

$$s_x^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2$$

How autocorrelated through time?

$$C(\tau) = \langle x_t x_{t+\tau} \rangle = \frac{1}{s_x^2 (N-\tau)} \sum_{t=1}^{N-\tau} (x_t - \bar{x})(x_{t+\tau} - \bar{x})$$

How predictable?

$$\Phi^m(r) = \frac{1}{N-m+1} \sum_{i=1}^{N-m+1} \ln C_r^m(i)$$

How stationary?

$$\frac{\mathrm{std}(\{\overline{x_{1:w}}, \overline{x_{w+1:2w}}, \ldots, \overline{x_{(m-1)w+1:mw}}\})}{\mathrm{std}(x)}$$

Fulcher (2018). Feature-based time-series analysis, *Feature Engineering,* CRC Press.

What feature(s) should I use?

Methods for time-series analysis have been developed across diverse scientific literature for decades
The *hctsa* feature set contains implementations of >7000 features, derived from hundreds of distinct methods.

Static Distribution

Quantiles
Trimmed means
Fits to standard distributions
Outliers
Moments
Entropy
Rank-orderings
Standard deviation

Correlation

Linear autocorrelation
Decay properties
Additive noise titration
Nonlinear autocorrelations
Time reversal asymmetry
Generalized self-correlation
Recurrence structure
Autocorrelation robustness
Scaling and fluctuation analysis
Permutation robustness
Local extrema
Seasonality tests
Zero crossing rates

Information Theory

Sample Entropy
Lempel-Ziv Complexity
Automutual information
Information dynamics
Approximate Entropy
Tsallis entropies

Model Fitting

Local prediction
GARCH models
Fourier fits
AR models
Exponential smoothing
State space models
Hidden Markov models
Piecewise splines
Biased walker simulations
ARMA models
Gaussian Processes

Stationarity

StatAv
Sliding window measures
Step detection
Distribution comparisons

Basis Functions

Wavelet transform
Peaks of power spectrum
Spectral measures
Power in frequency bands

(Phys) Nonlinear

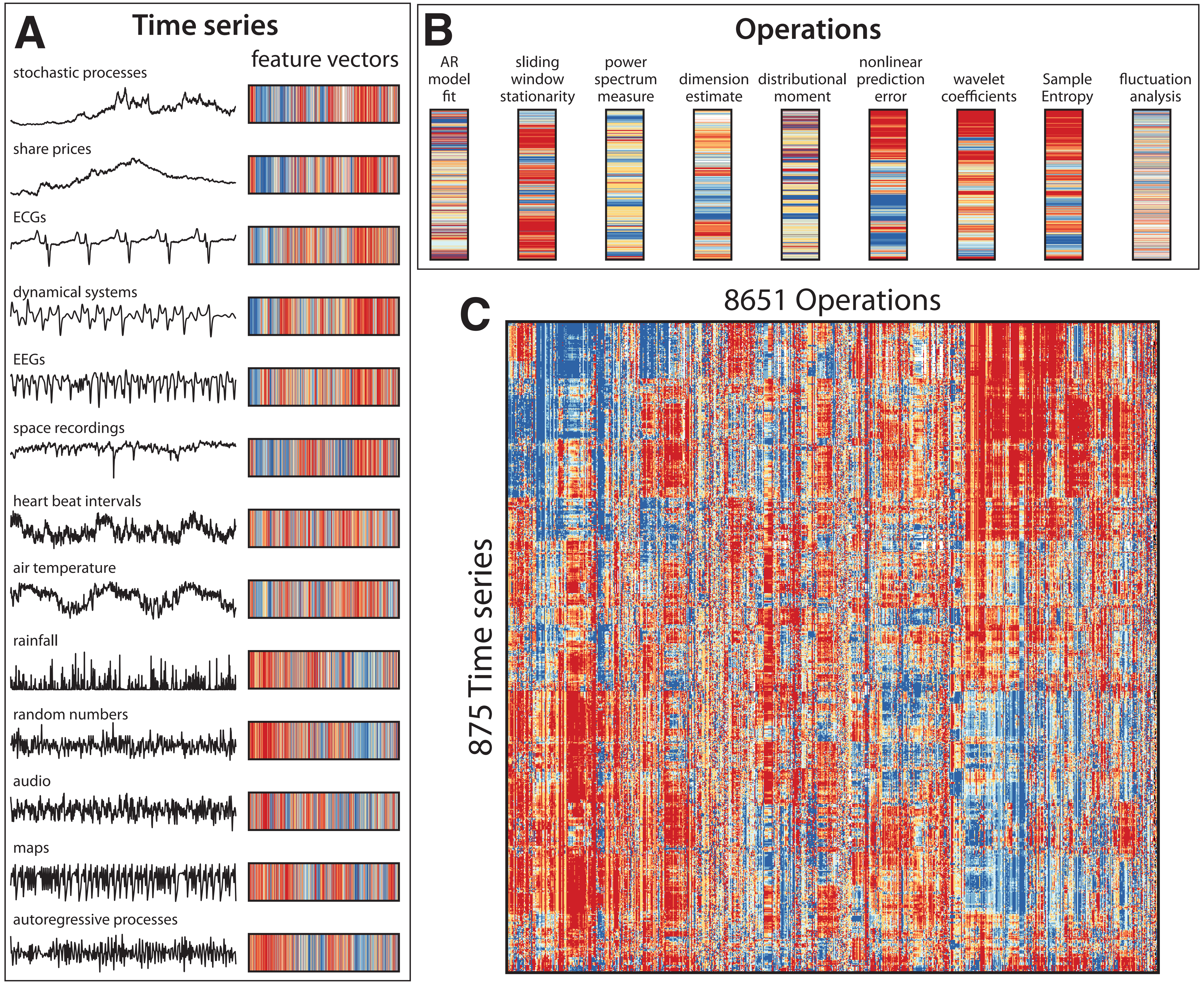
2D embedding structure
Taken's estimator
Fractal dimension
Correlation dimension
Poincaré sections
Surrogate data
Nonlinear prediction error
Lyapunov exponent estimate
False nearest neighbors

Others

Transition matrices
Local motifs
Dynamical system coupling
Visibility graph
Stick angle distribution
Extreme events
Singular spectrum analysis
Domain-specific techniques

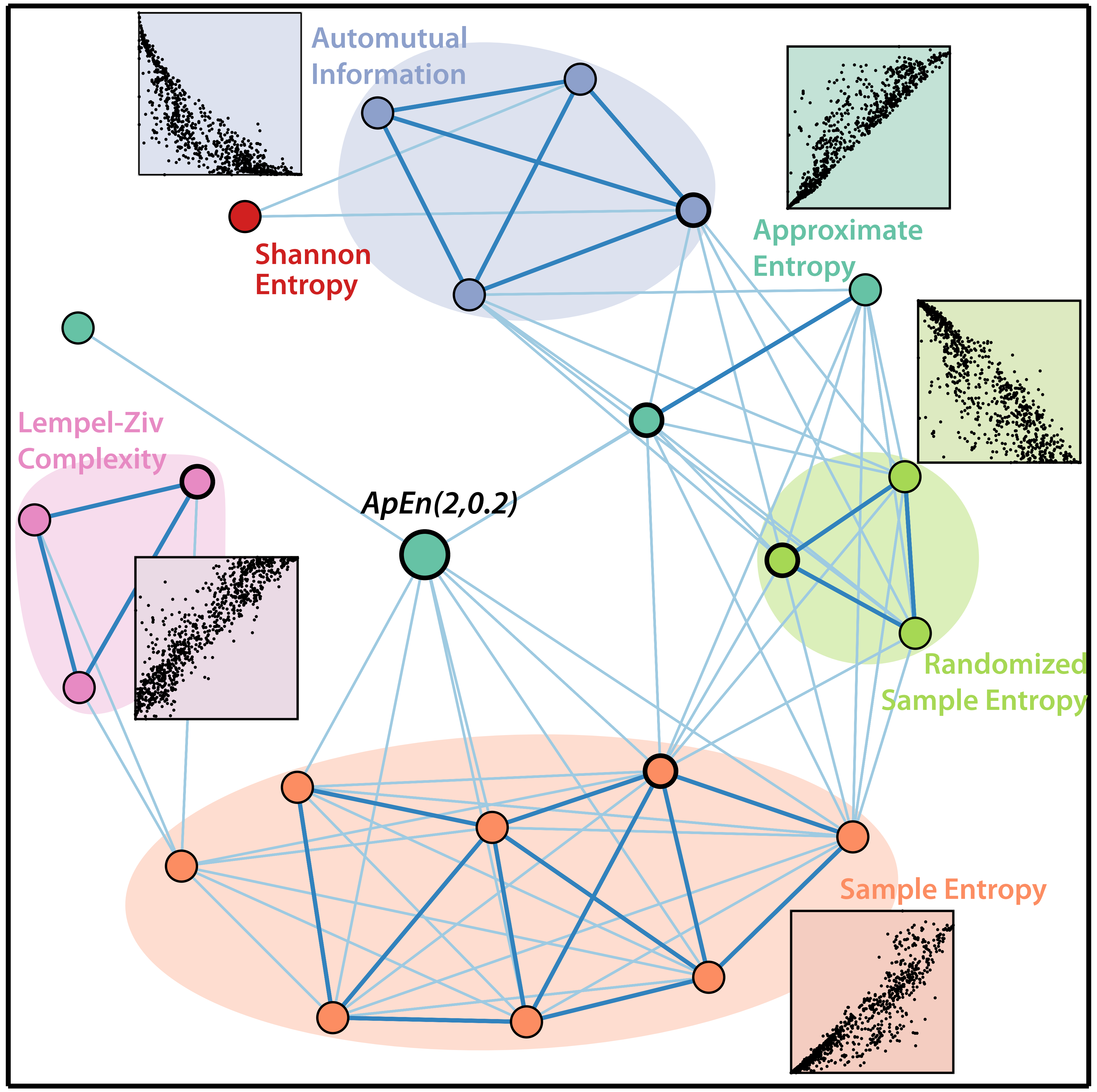
Structuring libraries of time-series data and analysis methods

We can represent a method by its behavior on data; a time series by its properties assessed by many scientific methods

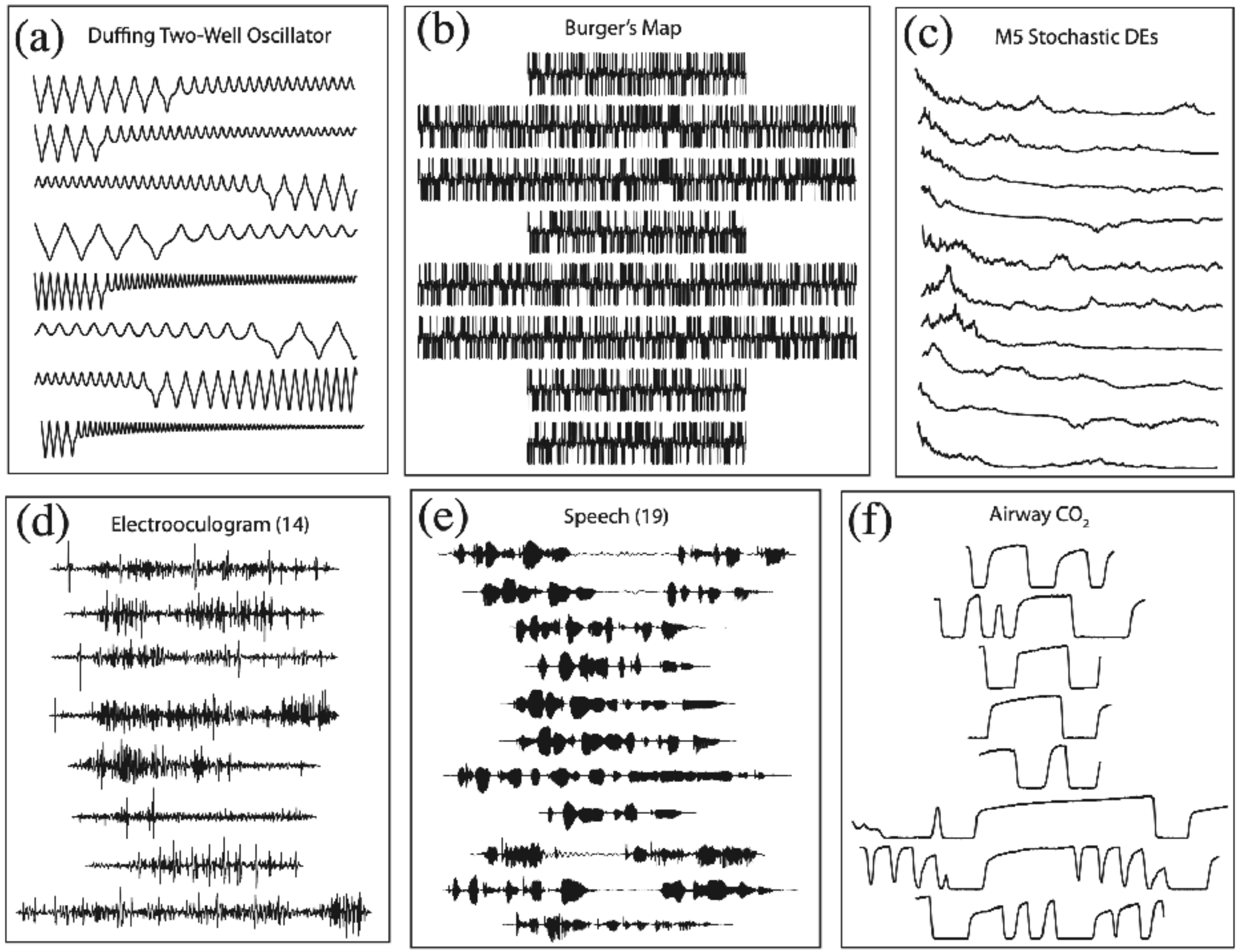


Fulcher et al. (2013). Highly comparative time-series analysis: the empirical structure of time series and their methods. *J. Roy. Soc. Interface*, **10:** 20130048.

We can organize diverse scientific methods for time-series analysis by their similarity in behavior on real data

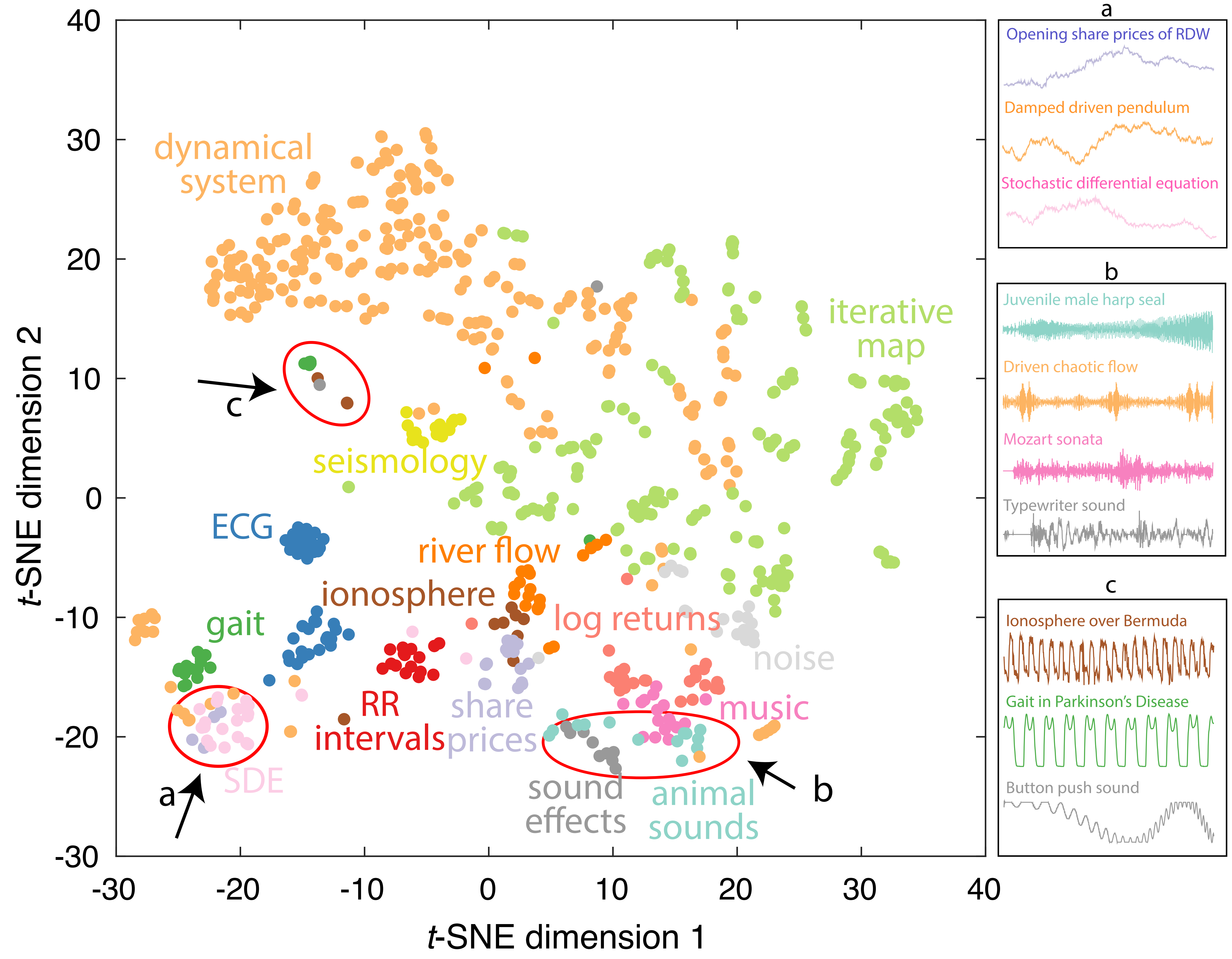


We can organize diverse time-series data based on the similarity of their properties



Low-dimensional feature-space projections

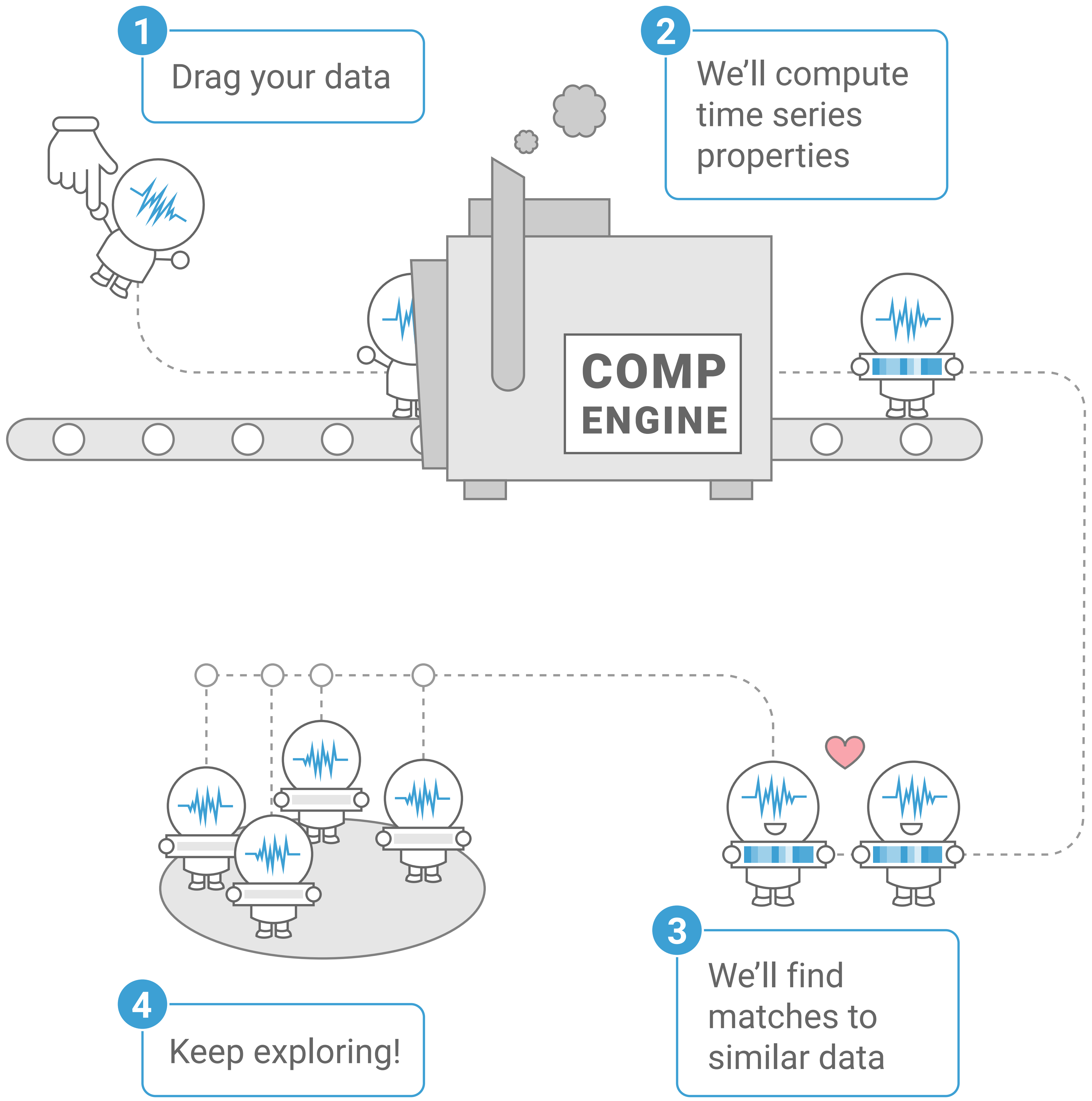
Projection of diverse data in a diverse, high-dimensional feature space



Fulcher et al. A self-organizing, living library of time-series data. *Scientific Data* **7**, 213 (2020).

Finding Connections

Are other scientists studying similar data to me?

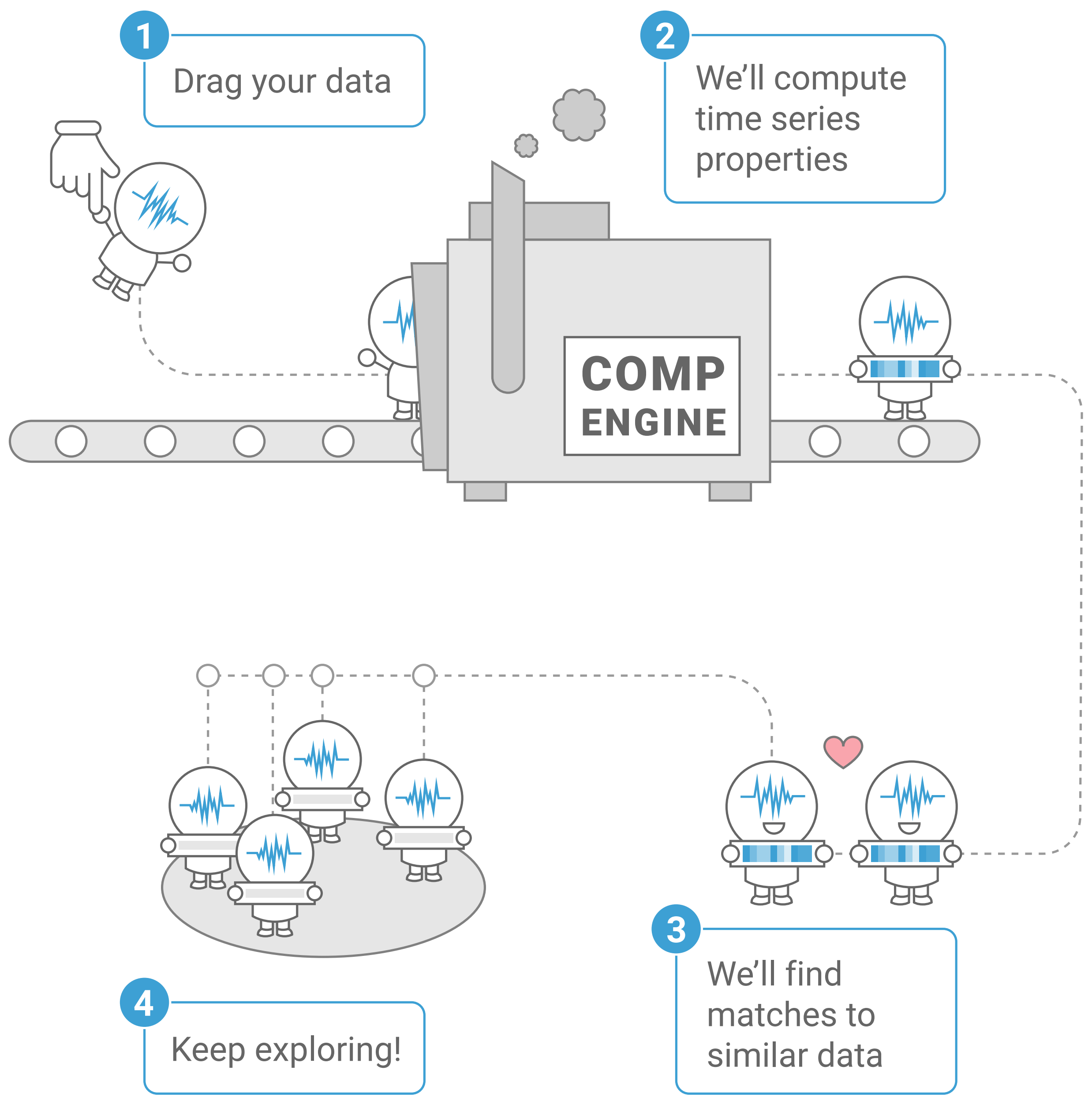


- *CompEngine Time Series* is a self-organizing database of interdisciplinary time-series data
- Connects diverse scientists through the structure of their data
- Bulk download functionality, and API for custom time-series data download: facilitates comprehensive empirical phenotyping of time-series analysis algorithms

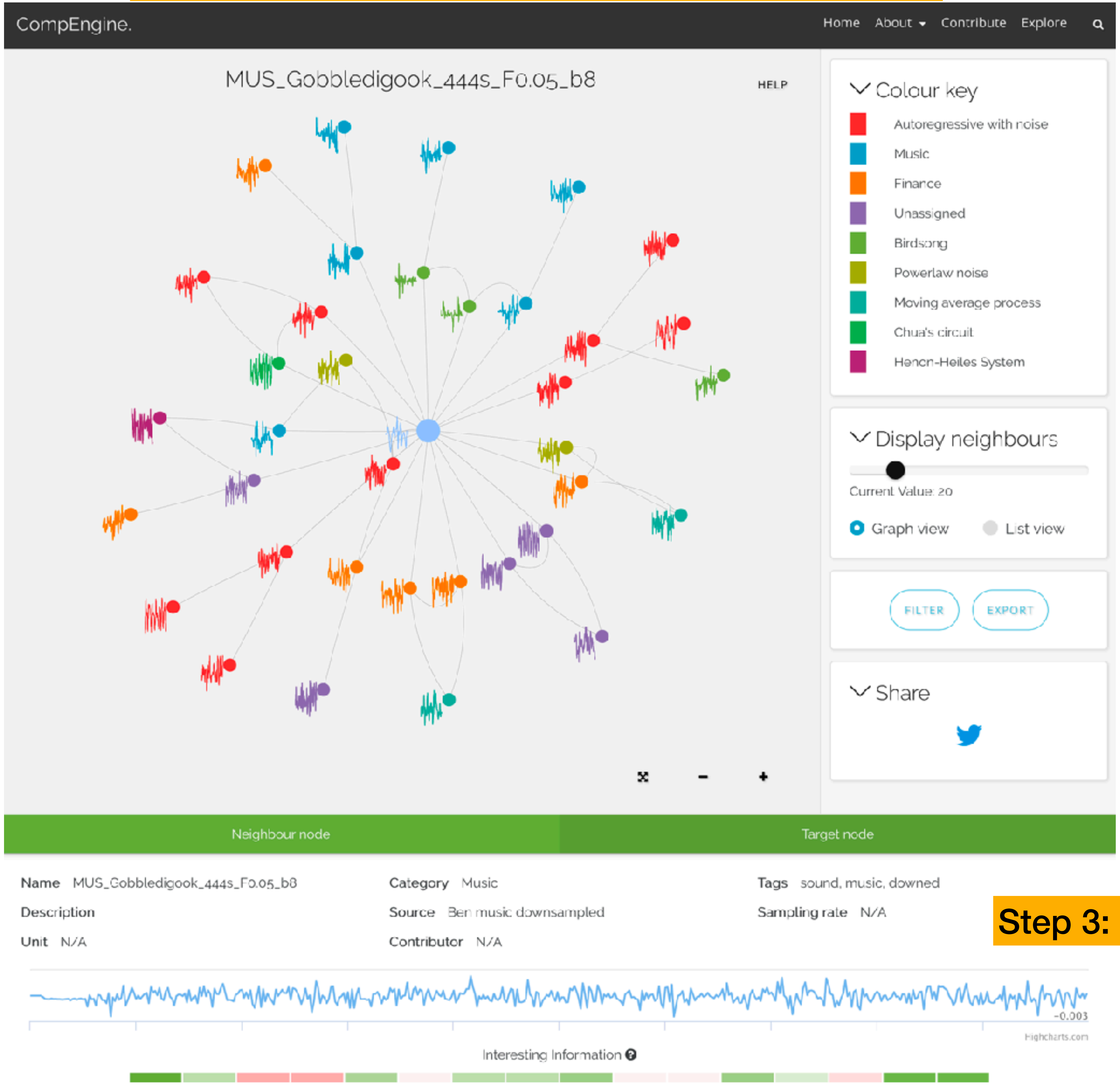
www.comp-engine.org

Fulcher et al. A Self-Organizing, Living Library of Time-Series Data. *Scientific Data* **7**: 213 (2020)

Step 1: Drag on your data



Step 2: Interactively Explore Similar Scientific Data

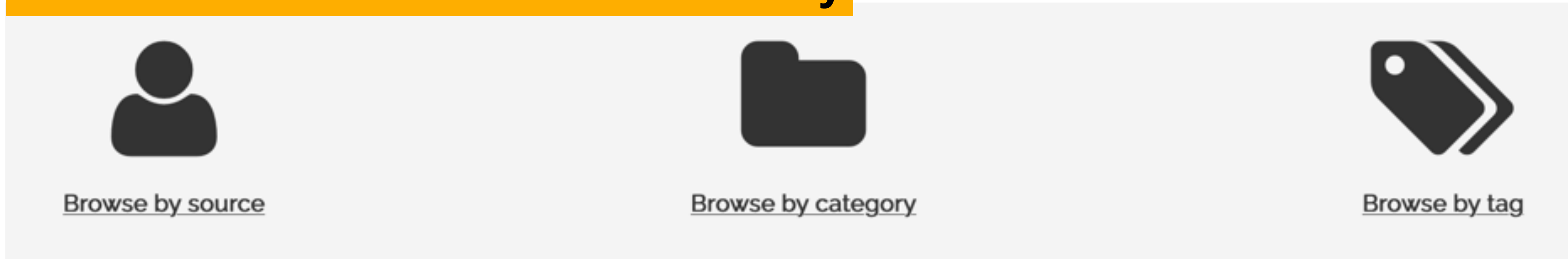


Demo

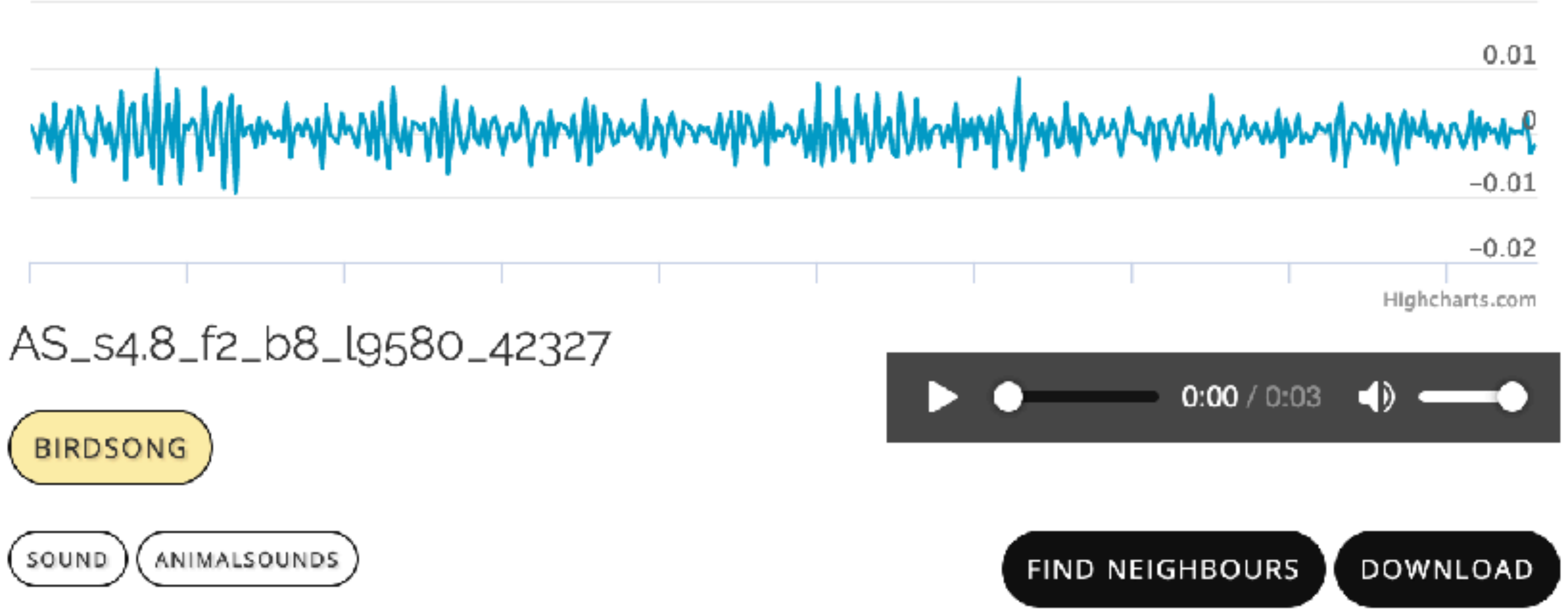
Step 3: Contribute your data

If you don’t have data on-hand, you can still explore

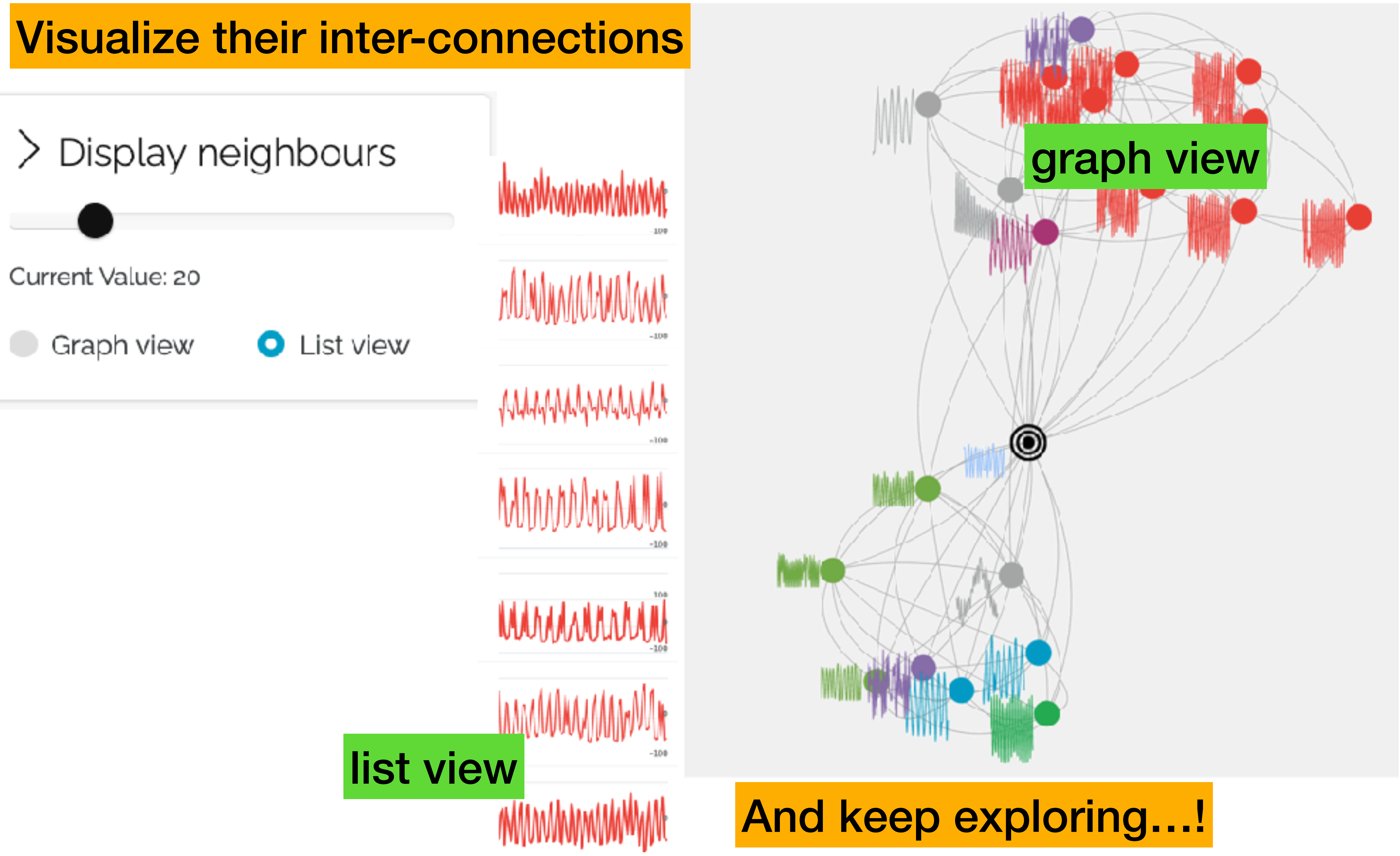
Browse the full time-series library



Interactively Explore Scientific Time Series

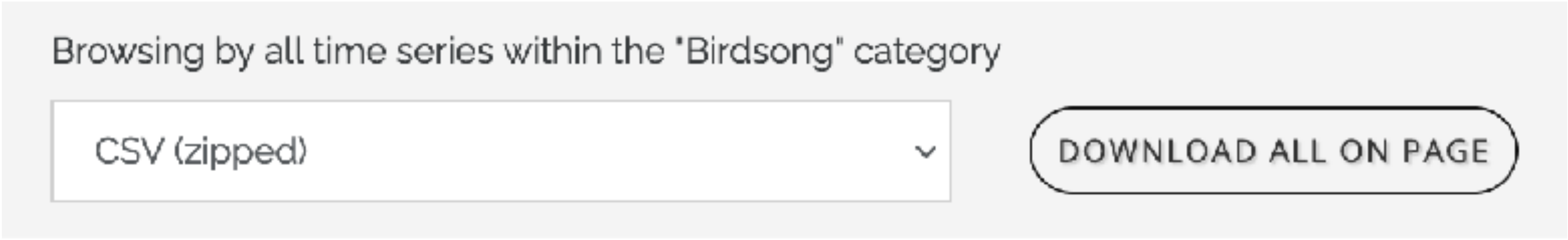


Visualize their inter-connections



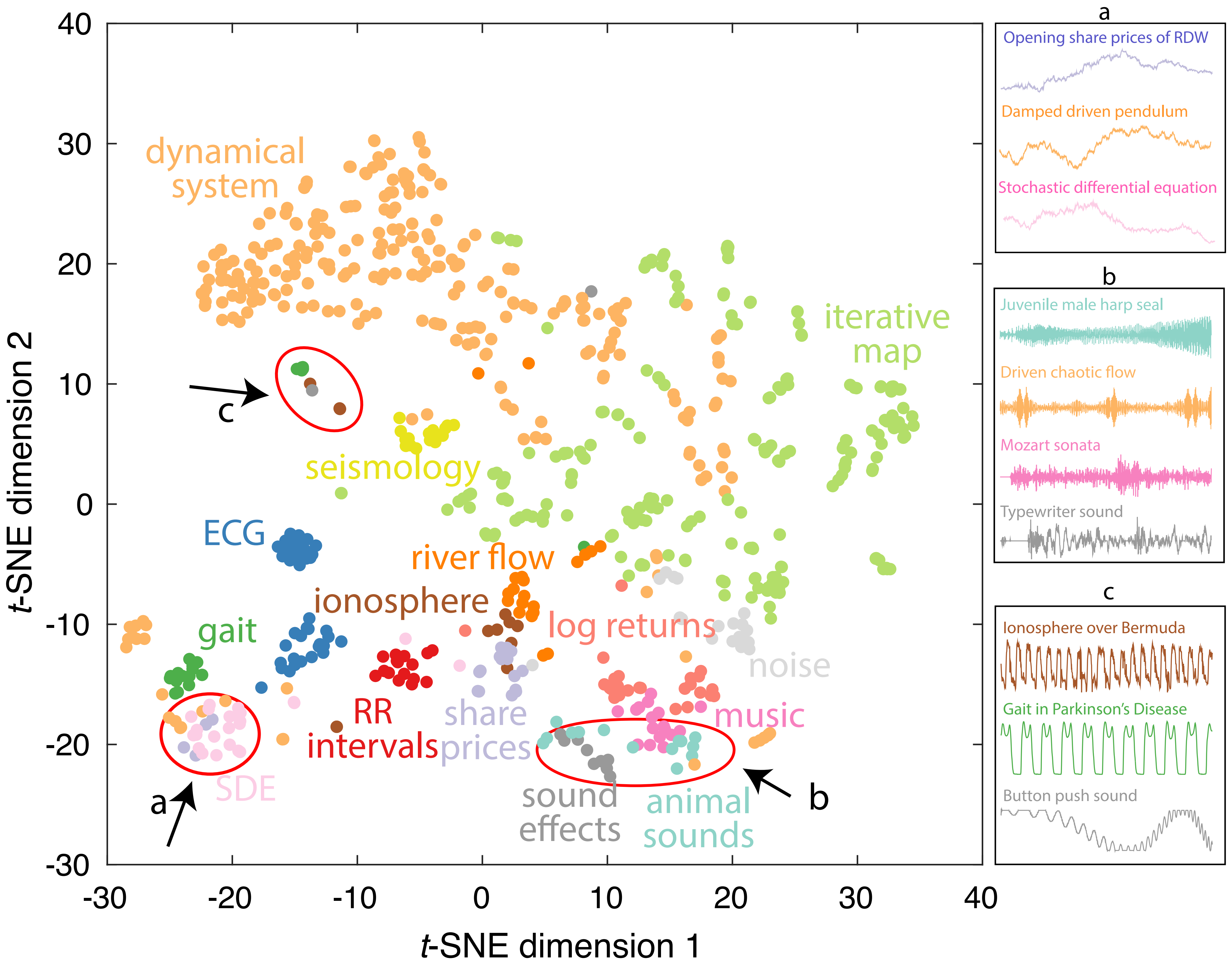
And keep exploring…!

Download any/all data you find:



A self-organizing database

Connects scientists through the data they analyze

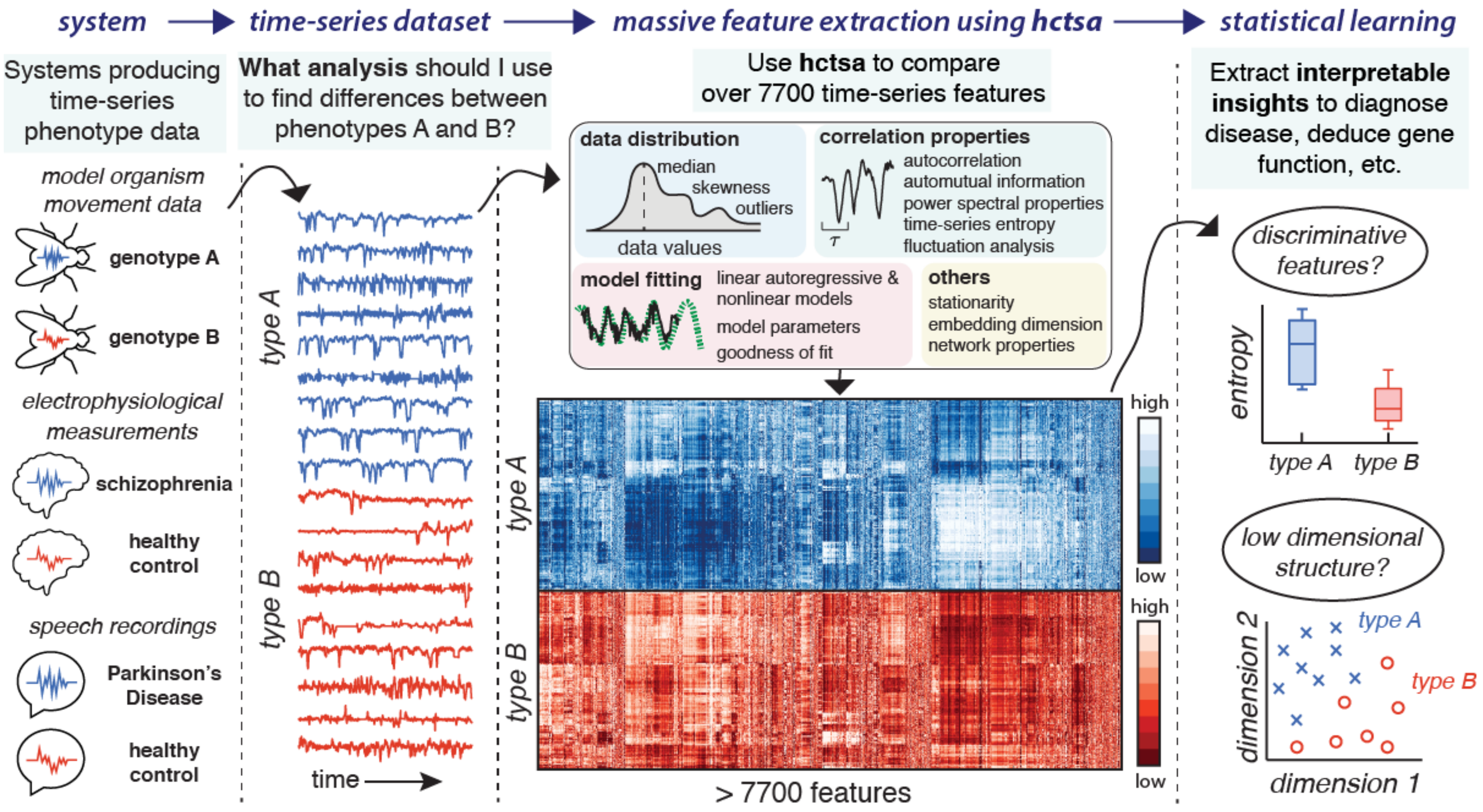


- Most (all?) databases are organized based on metadata, not extracted properties.
- *Connects scientists through their data*: may overcome barriers to connecting scientists with different expertise for meaningful collaboration
- A comprehensively library of data with which to assess the performance/behavior of analysis methods (strengths & weaknesses).

Fulcher et al. A Self-Organizing, Living Library of Time-Series Data. *Scientific Data* **7**: 213 (2020).

The highly comparative approach

Compare the performance of a comprehensive library of scientific time-series methods: pick those that best suit your problem

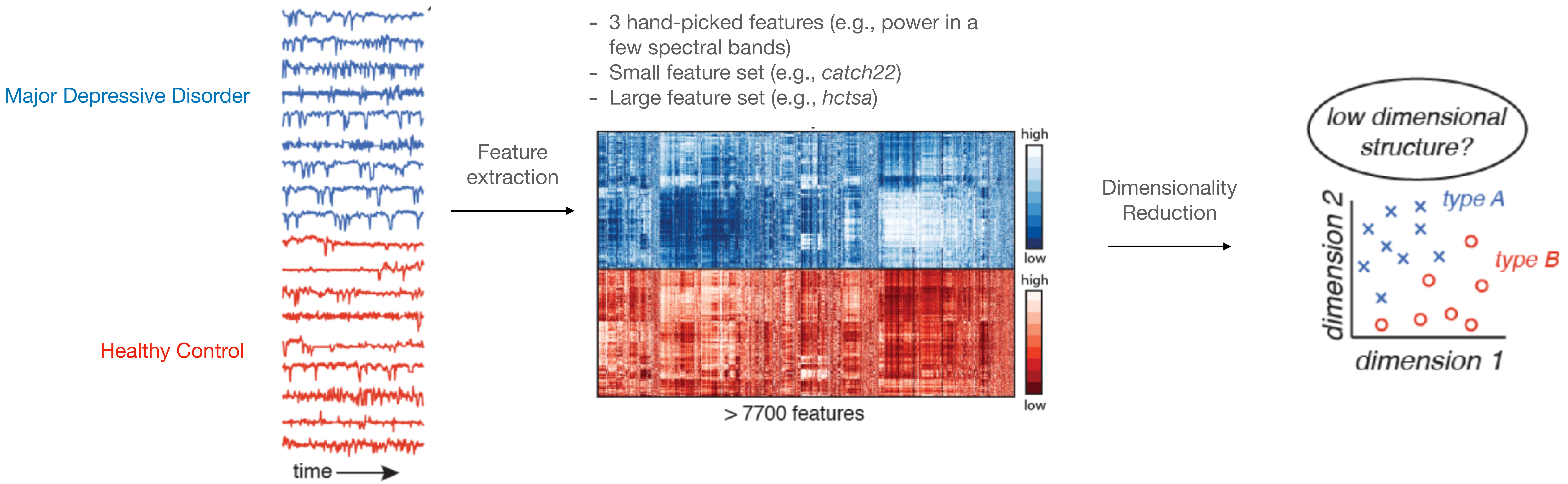


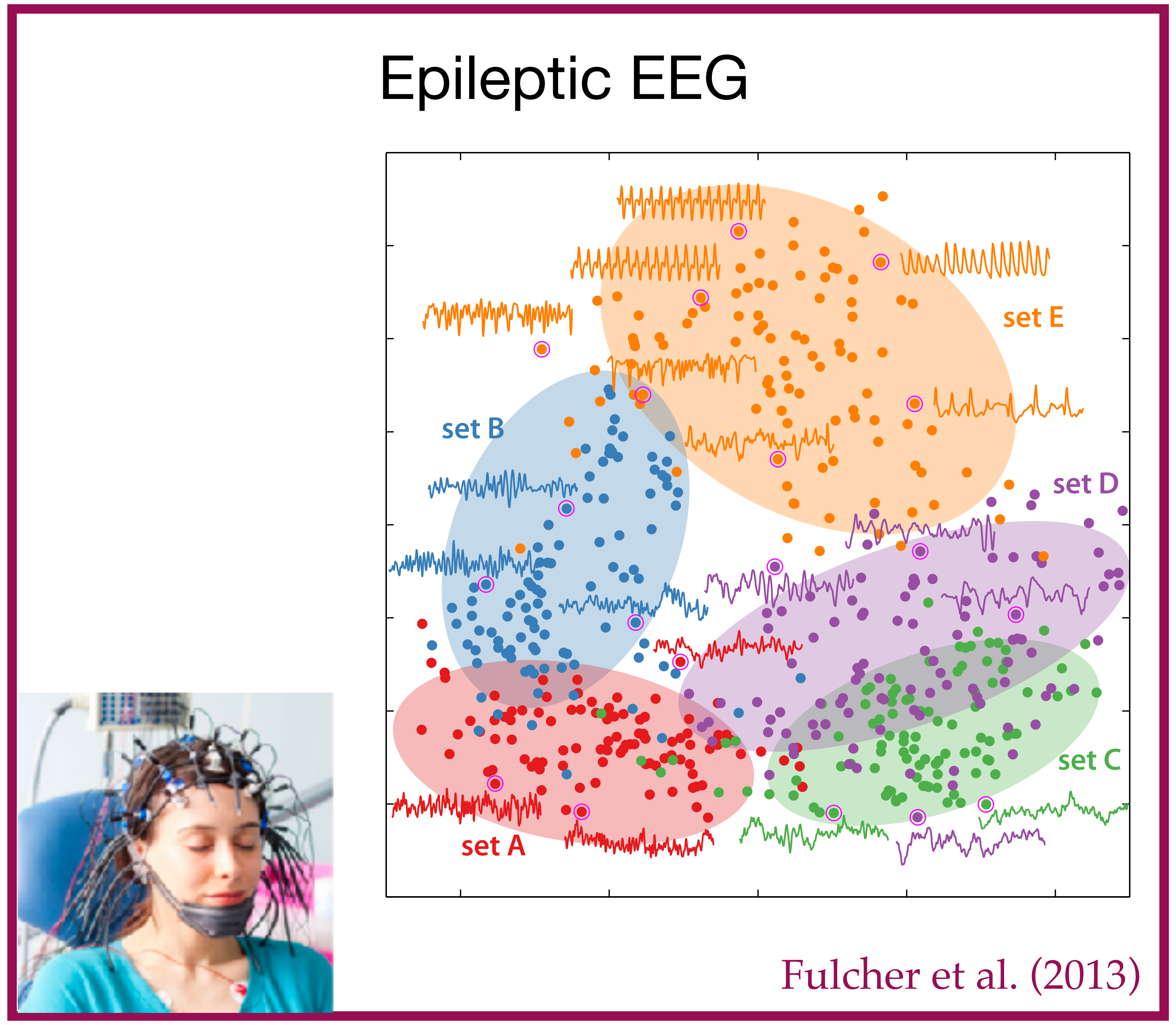
Fulcher & Jones (2017). *hctsa*: A Computational Framework for Automated Time-Series Phenotyping Using Massive Feature Extraction. *Cell Systems*, **5**, 527–531.

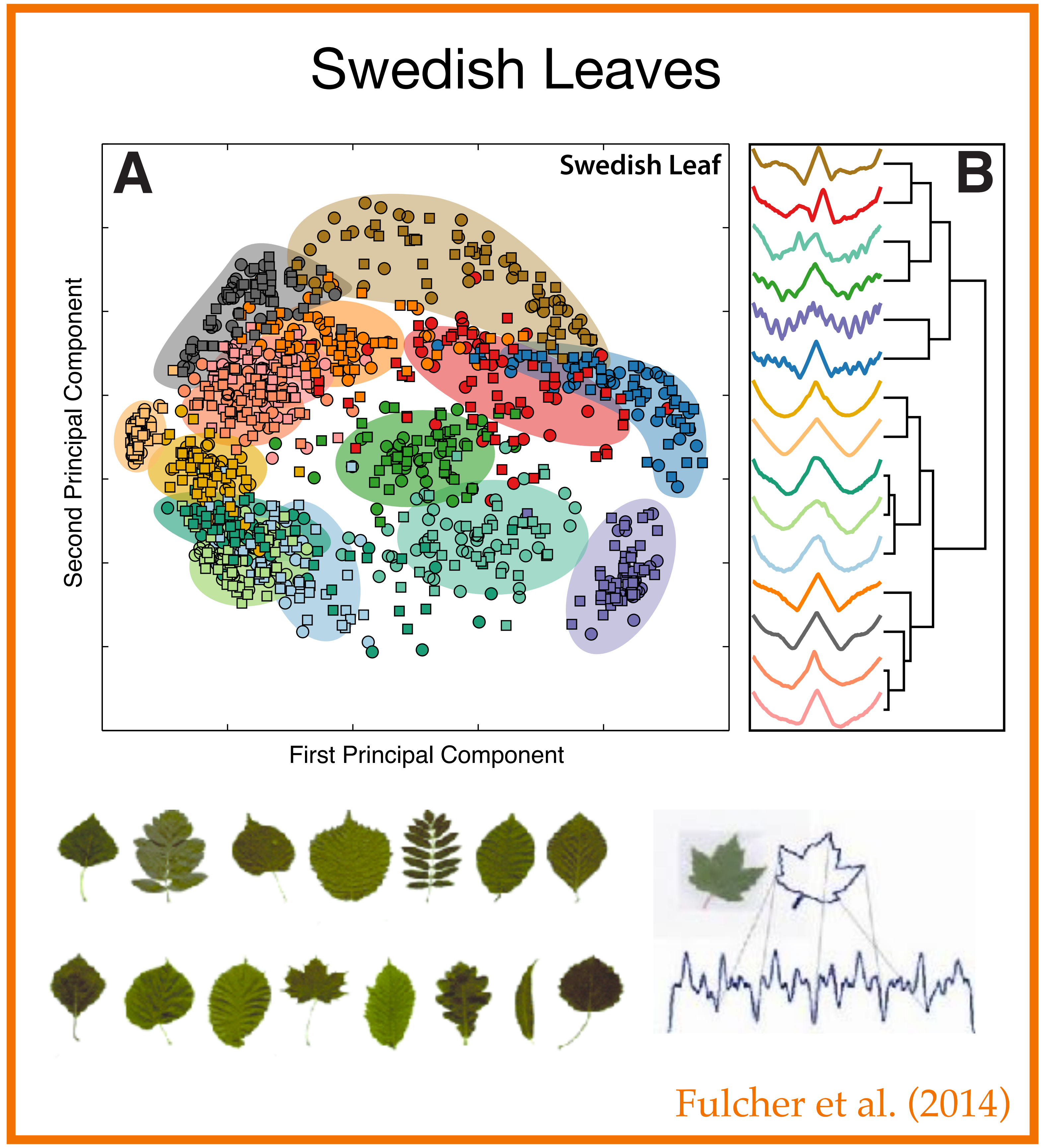
Low-Dimensional Feature-Space Projections

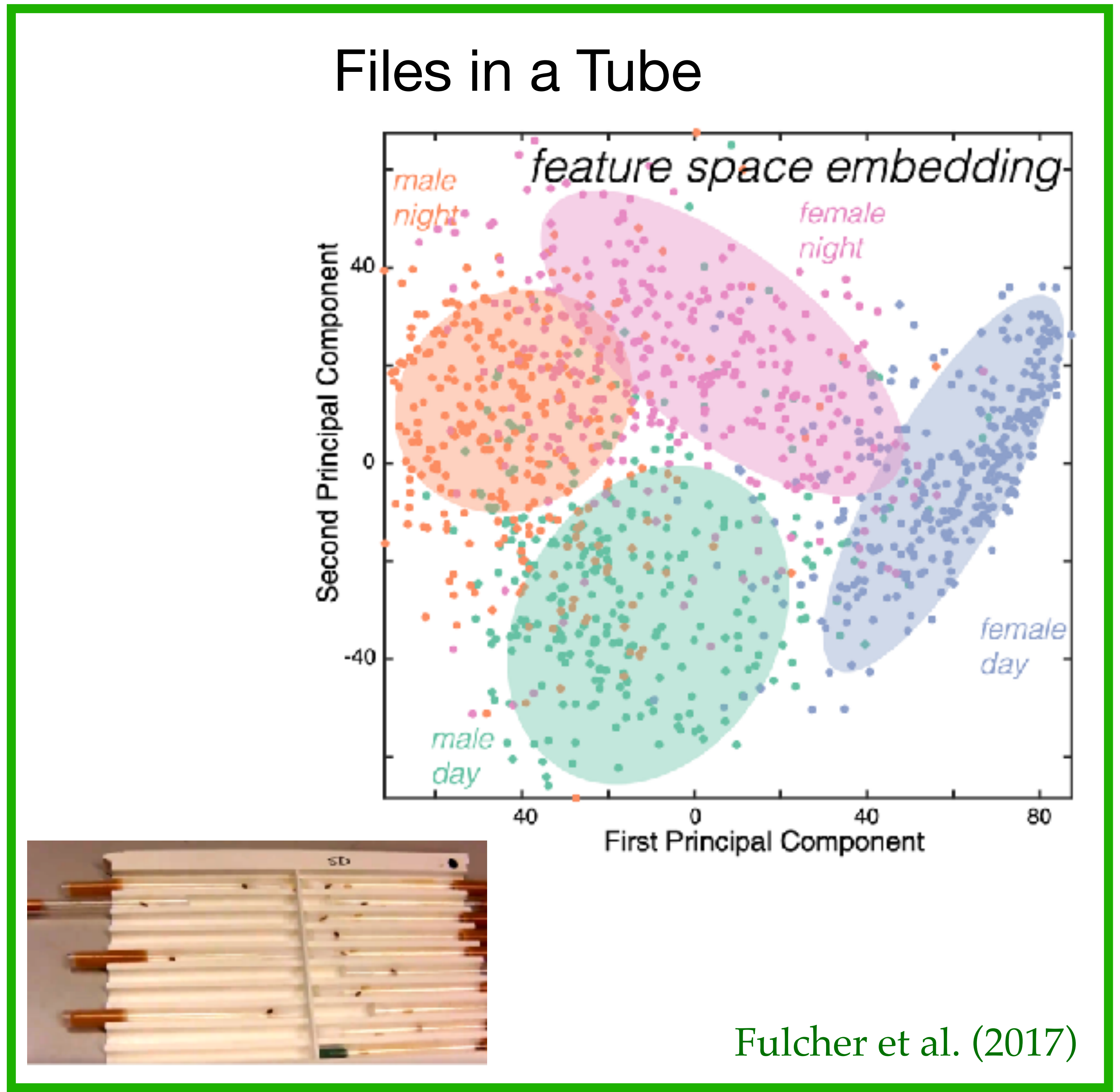
How are my time-series data structured?

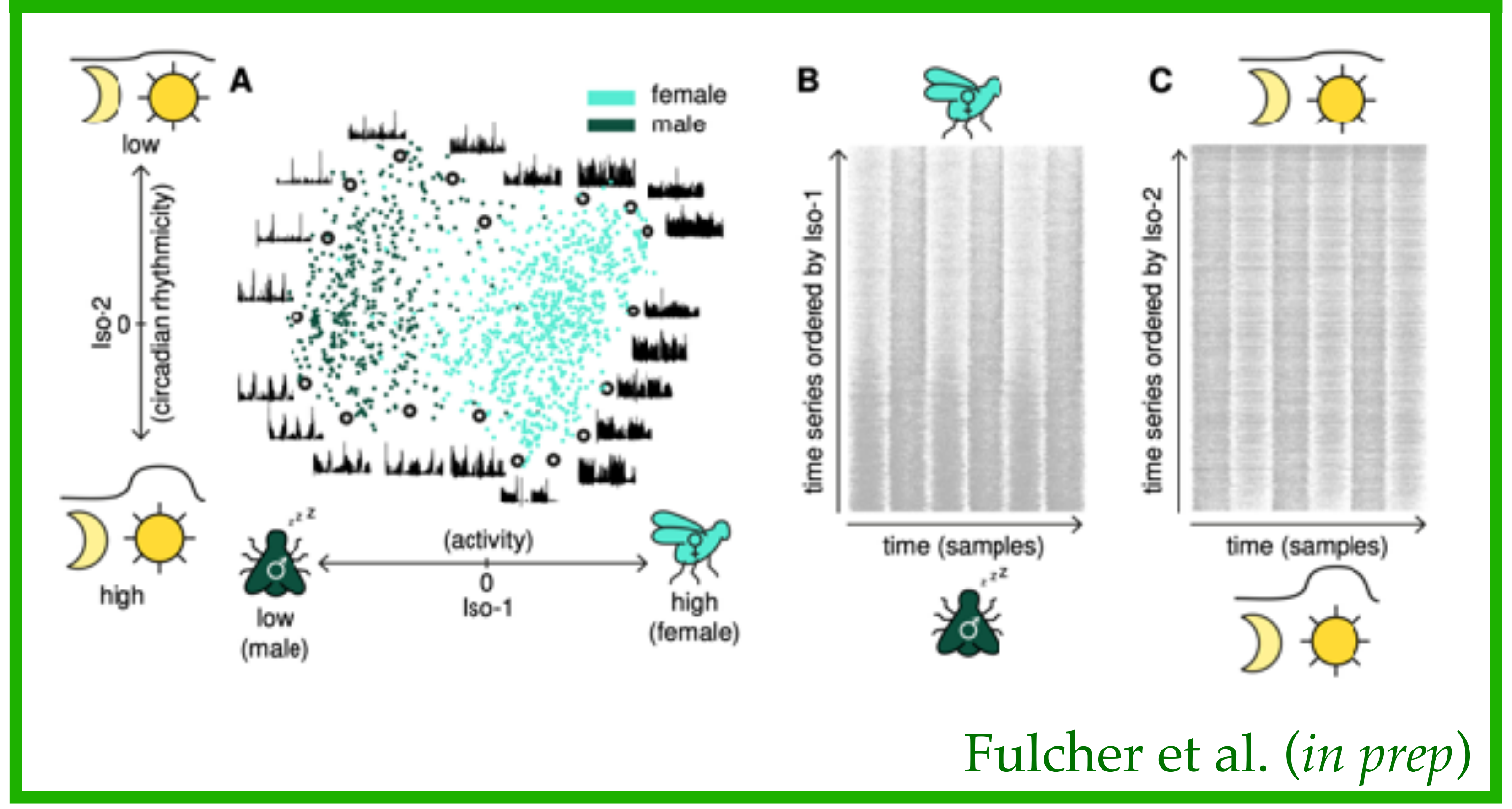
Represent each time series as a set of features (interpretable structural properties), and look for patterns in the low-dimensional feature space: ***time series with similar properties are close in the space.***

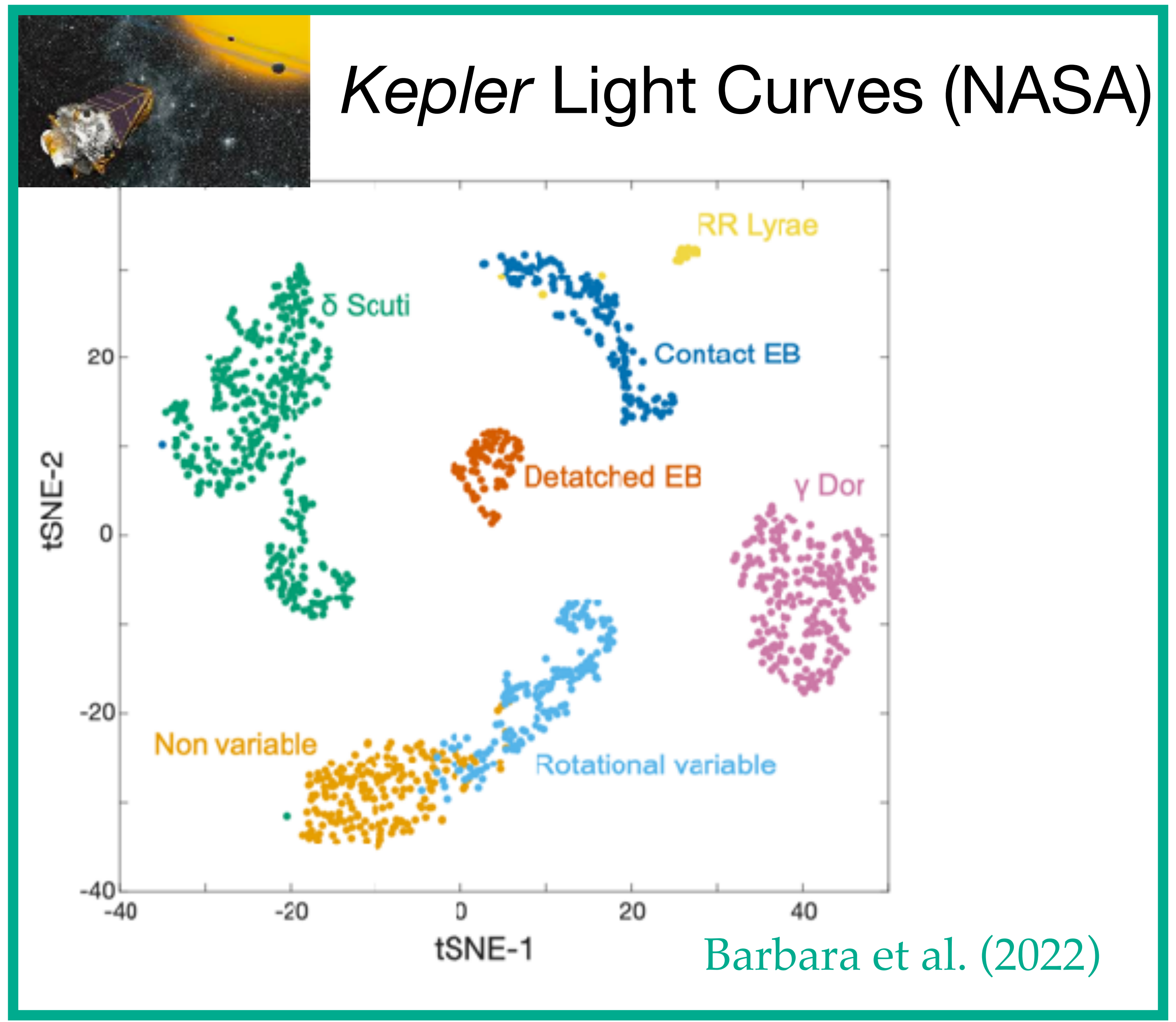


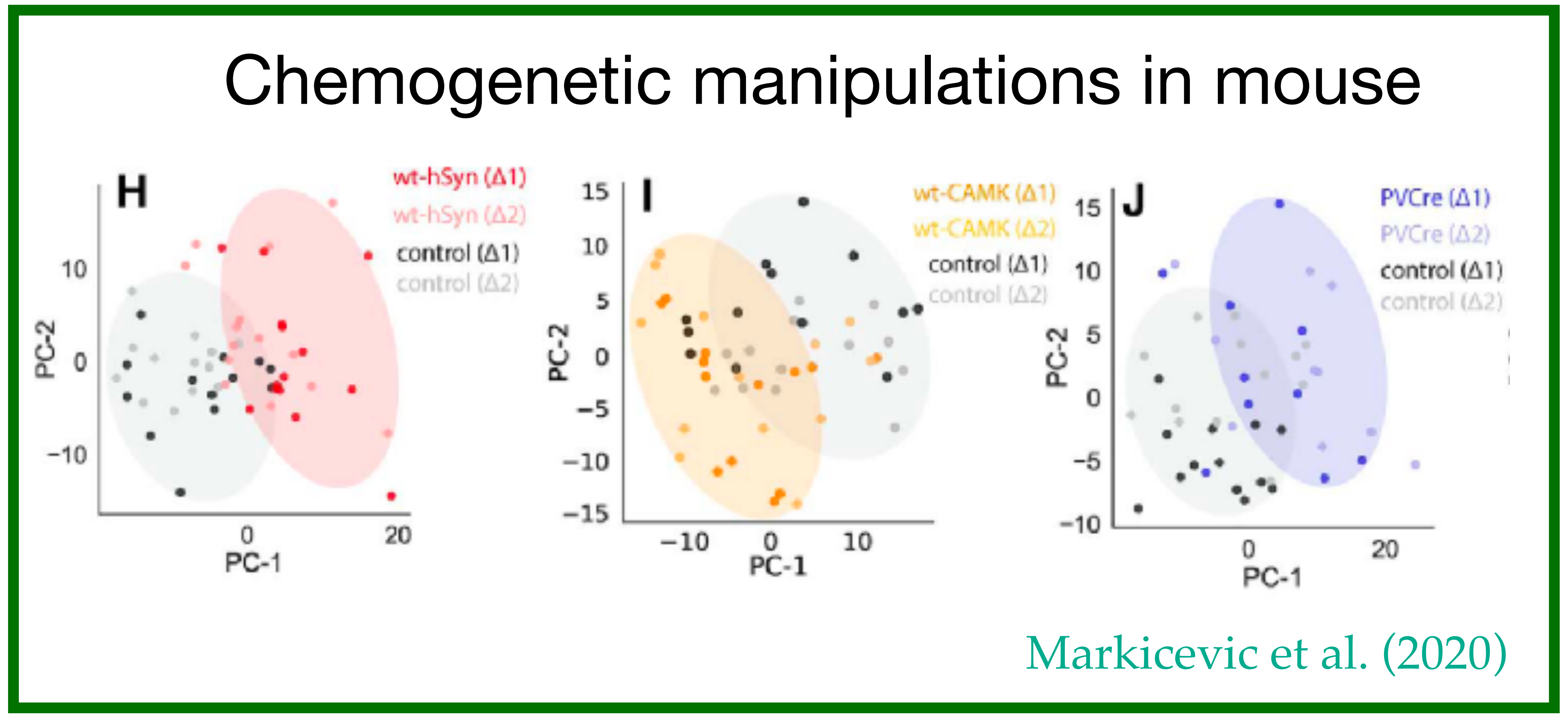

Epileptic EEG
set E
set B
set D
set C
set A
Fulcher et al. (2013)


Swedish Leaves
A
Swedish Leaf
B
Second Principal Component
First Principal Component
Fulcher et al. (2014)


Files in a Tube
feature space embedding
male night
female night
female day
male day
Second Principal Component
First Principal Component
40
0
-40
80
Fulcher et al. (2017)


A
B
C
female
male
low
high
Iso-2 (circadian rhythmicity)
(activity)
Iso-1
low (male)
high (female)
time series ordered by Iso-1
time series ordered by Iso-2
time (samples)
Fulcher et al. (in prep)


Kepler Light Curves (NASA)
RR Lyrae
δ Scuti
Contact EB
Detatched EB
γ Dor
Non variable
Rotational variable
tSNE-2
tSNE-1
Barbara et al. (2022)


Chemogenetic manipulations in mouse
H
I
J
wt-hSyn (Δ1)
wt-hSyn (Δ2)
control (Δ1)
control (Δ2)
wt-CAMK (Δ1)
wt-CAMK (Δ2)
PVCre (Δ1)
PVCre (Δ2)
PC-1
PC-2
Markicevic et al. (2020)

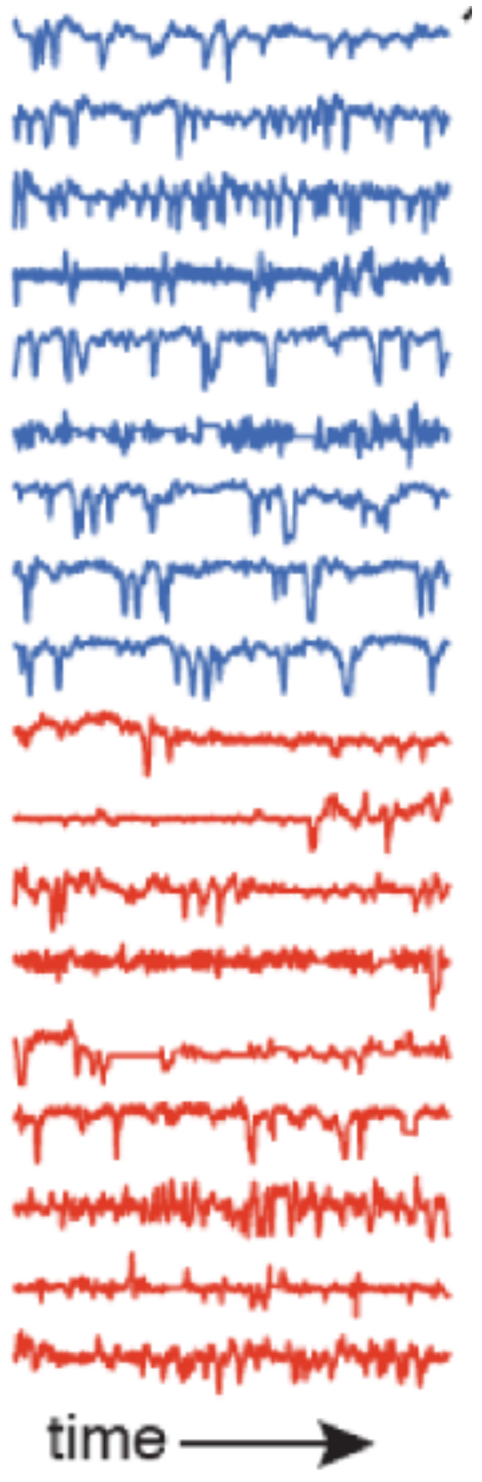
Classification

What types of features distinguish classes in my dataset?

(straightforward extension to real-valued labels: regression)

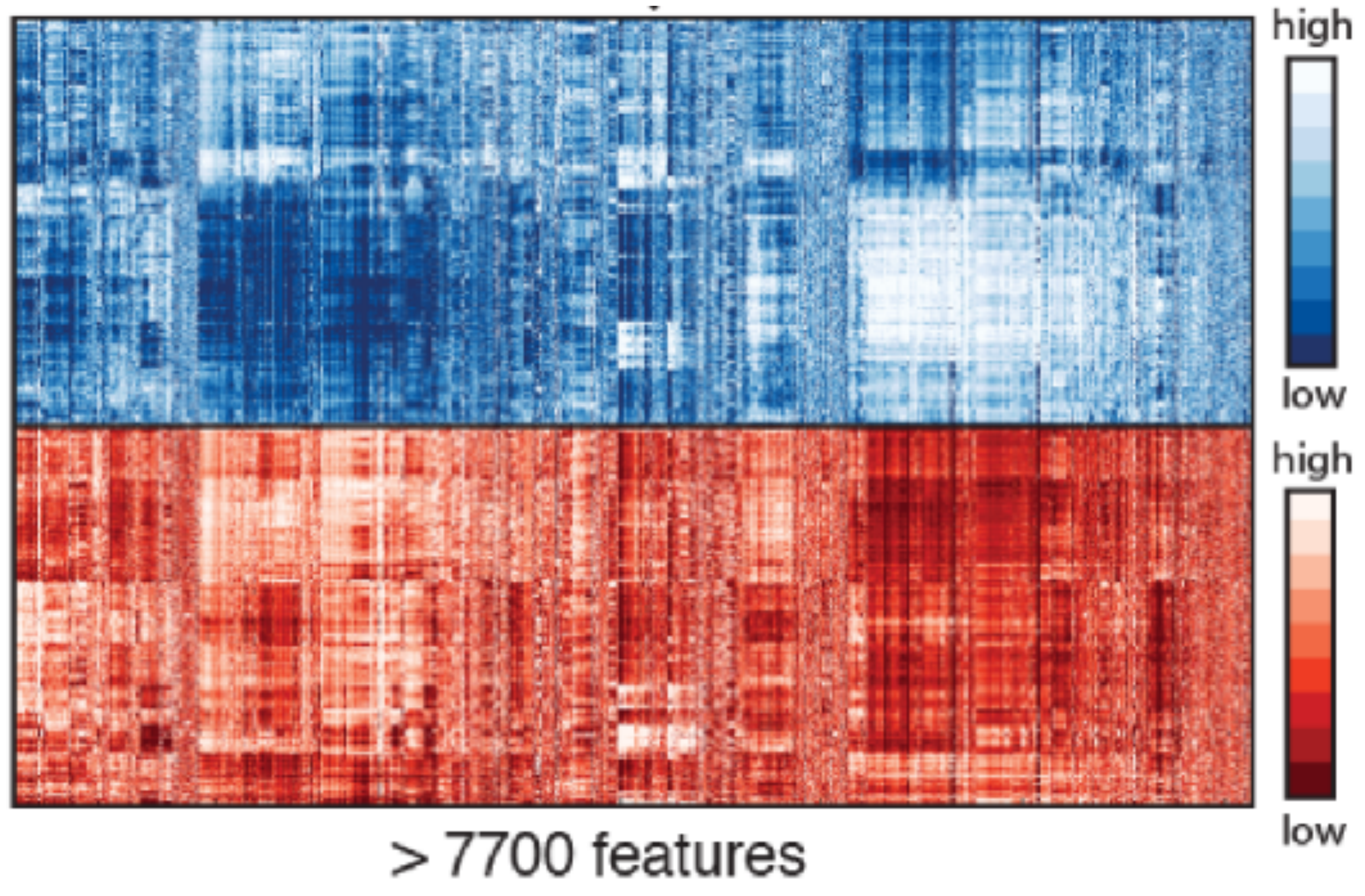
Major Depressive Disorder

Healthy Control



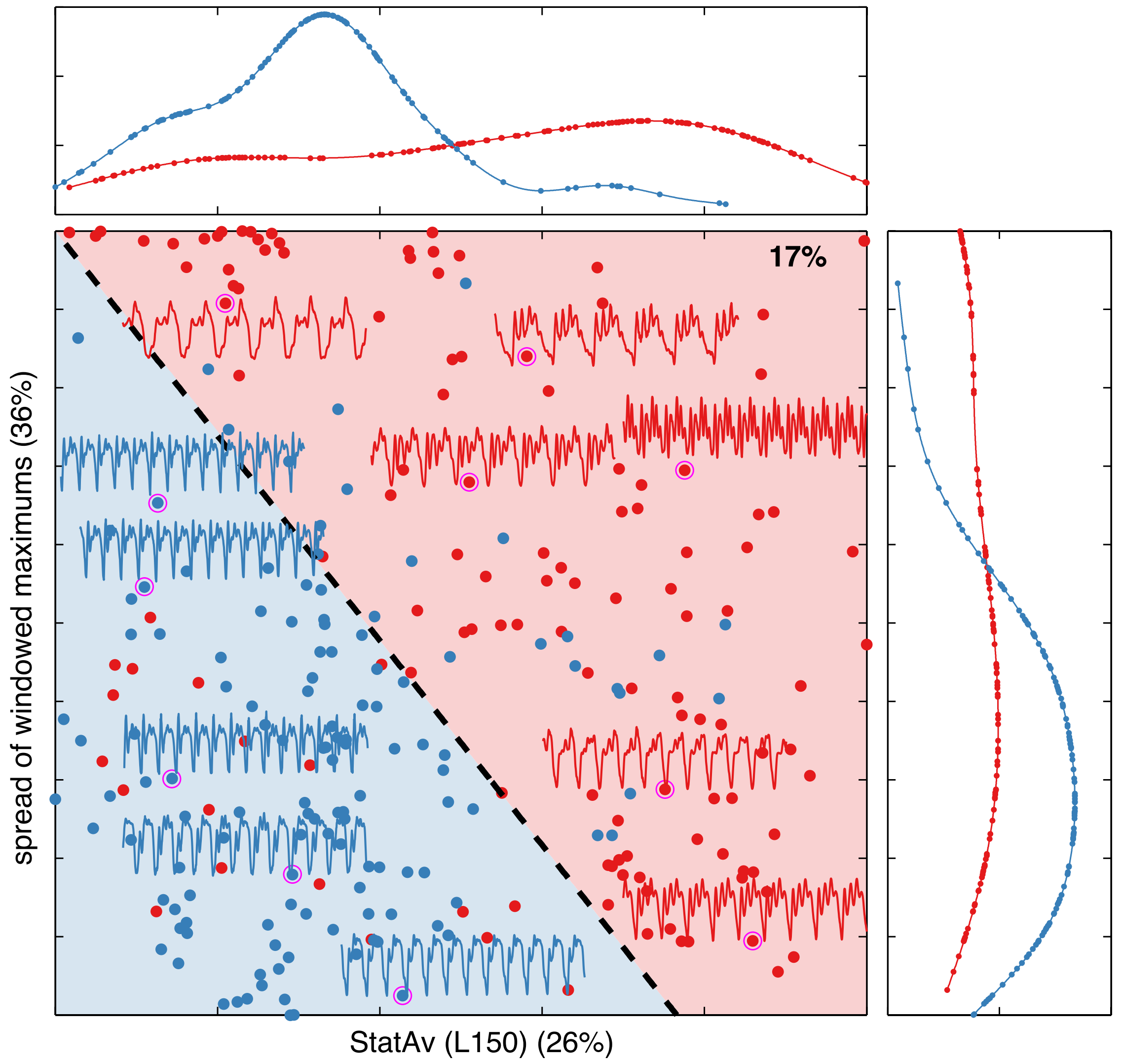
Feature extraction

- 3 hand-picked features (e.g., power in a few spectral bands)
- Small feature set (e.g., *catch22*)
- Large feature set (e.g., *hctsa*)



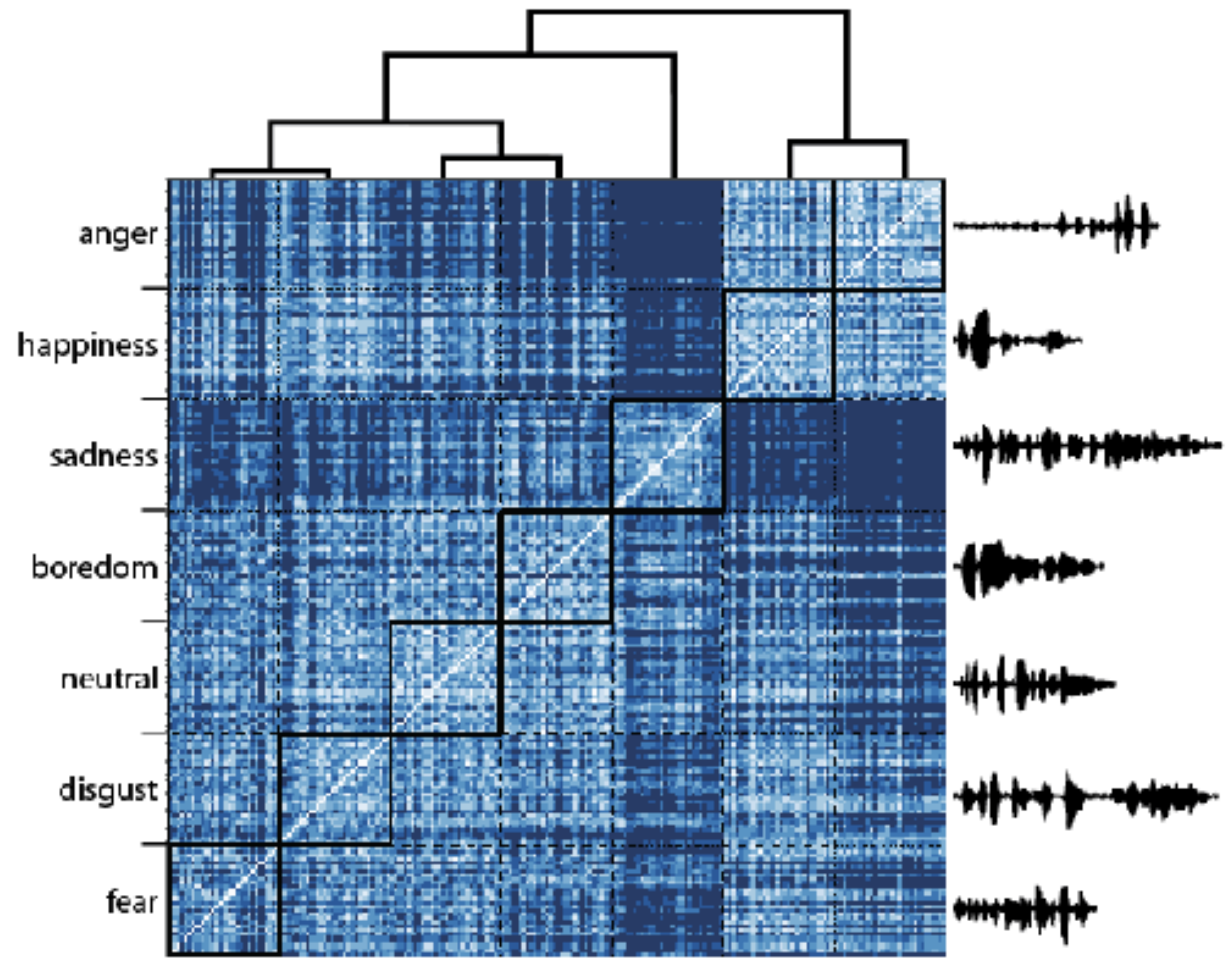
Train classifier

Classifying Parkinsonian Speech



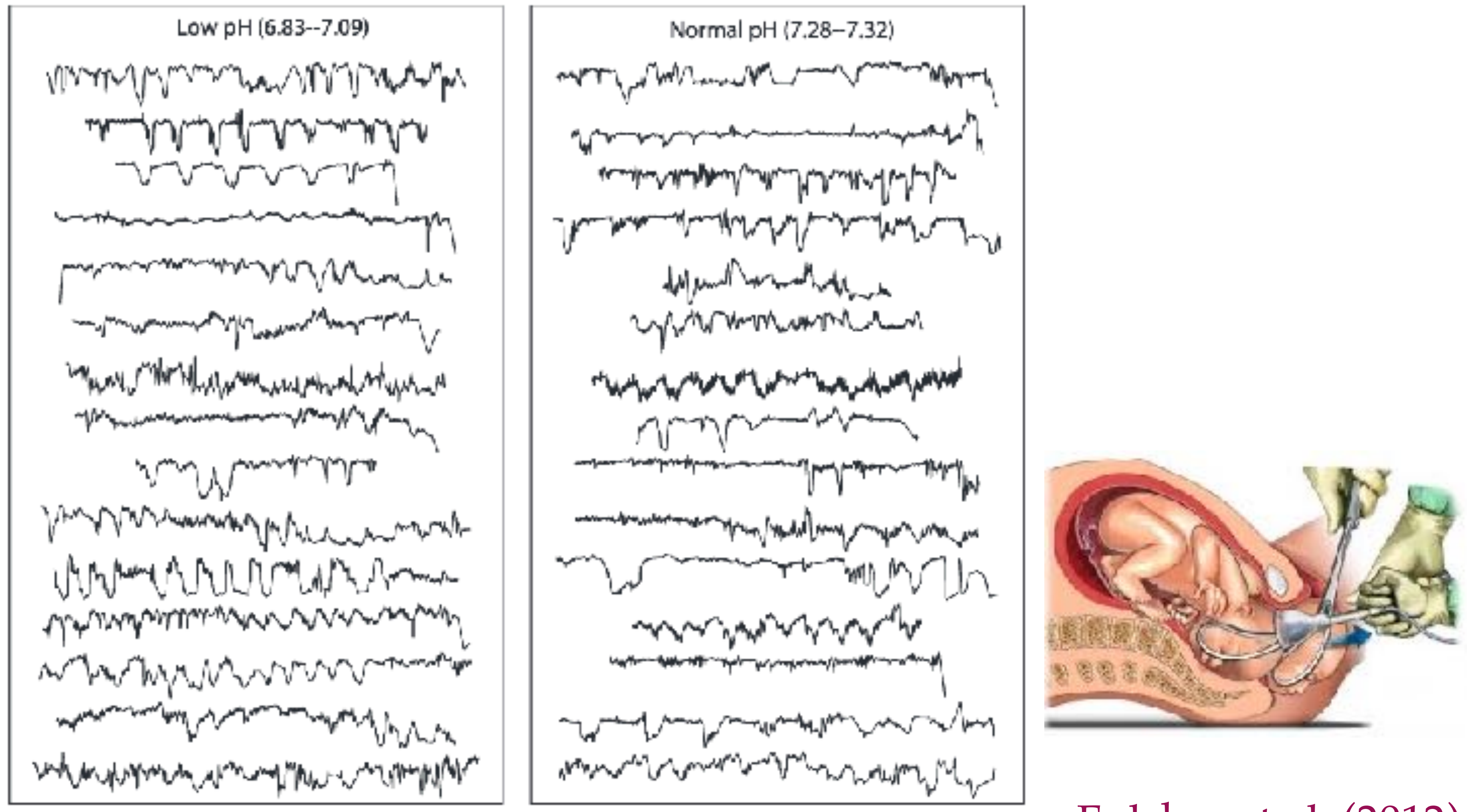
Fulcher et al. (2013)

Classifying Emotions from Speech



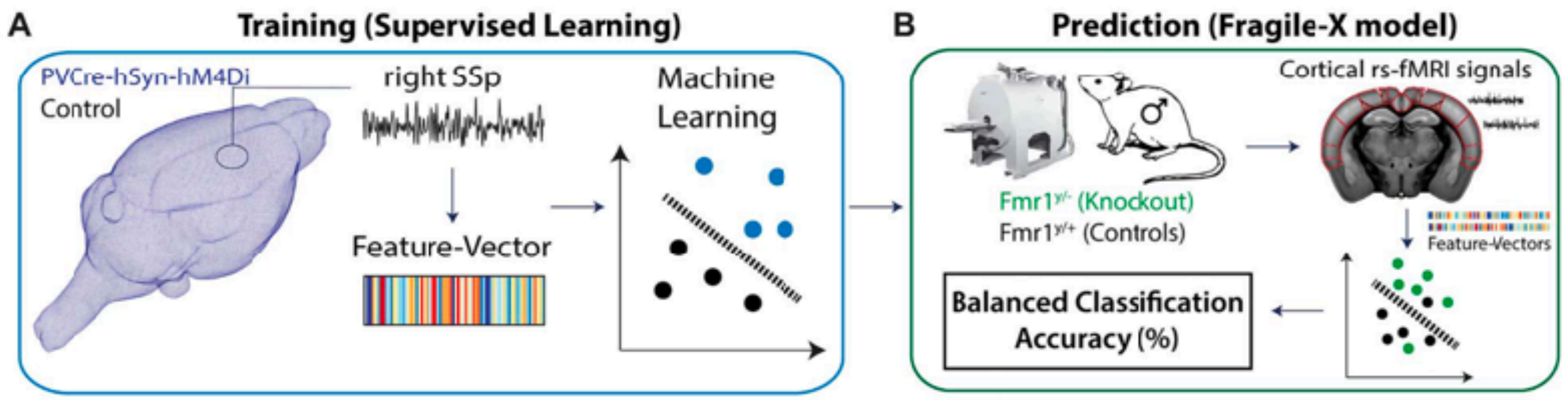
Fulcher et al. (2013)

Data-Driven Labour Interventions



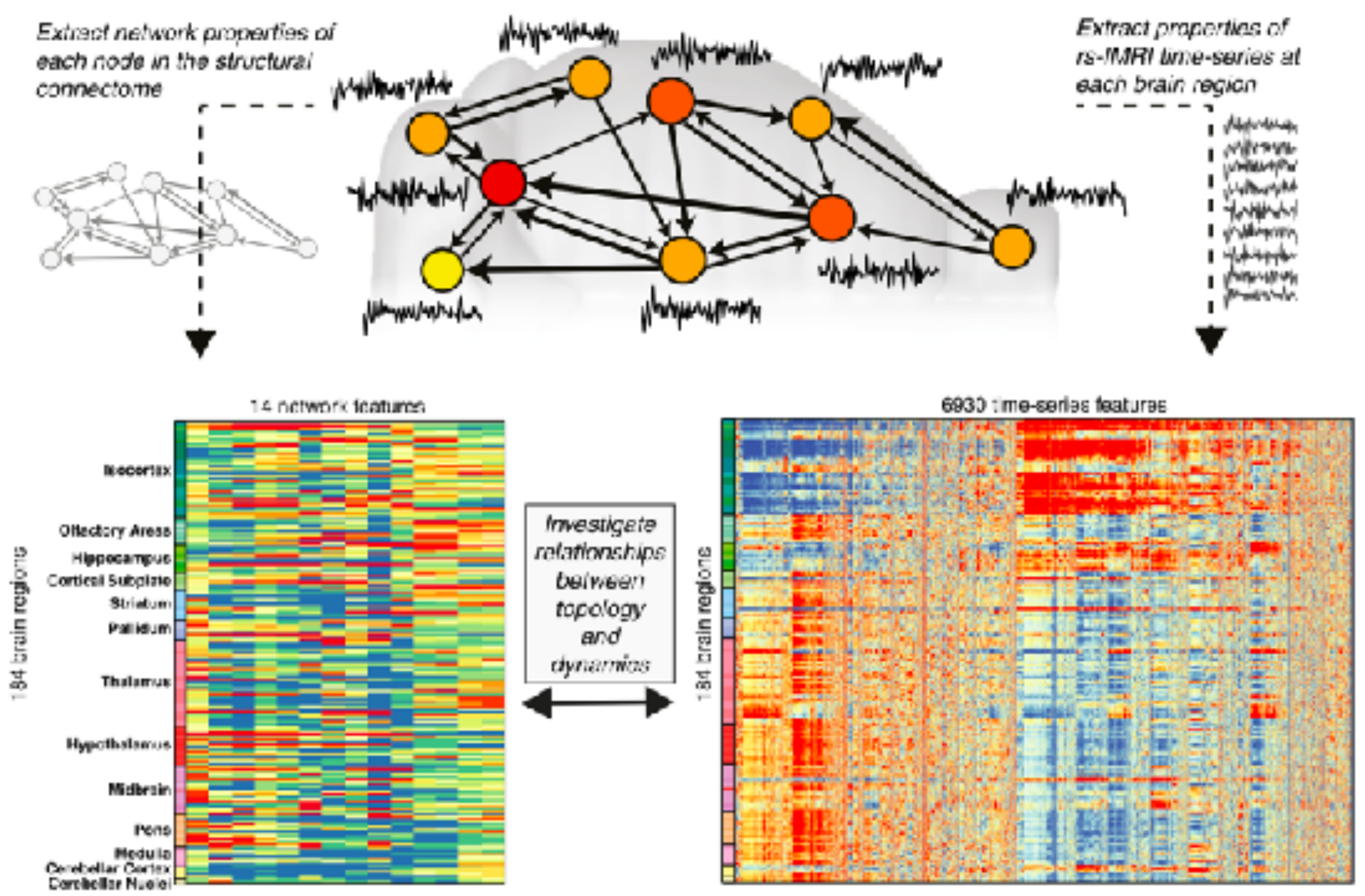
Fulcher et al. (2012)

Chemogenetic manipulations for mouse fMRI



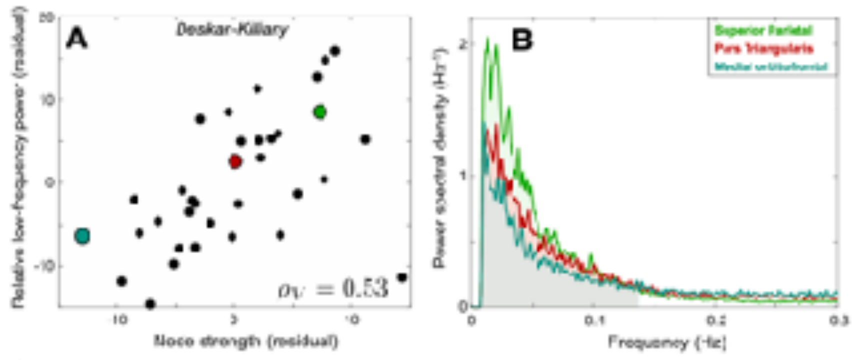
Markicevic et al. (2020).

Structure-function coupling in mouse



Sethi et al. (2017)

…and human



Fallon et al. (2020)

https://github.com/benfulcher/hctsa/wiki/Publications-using-hctsa

Distinguishing types of energy use in buildings

Liu et al. (2019). A hybrid model for appliance classification based on time series features. *Energy and Buildings*, **196**, 112-123.

Miller, C. (2019). What's in the box?! Towards explainable machine learning applied to non-residential building smart meter classification. *Energy and Buildings*, **199**, 523-536.

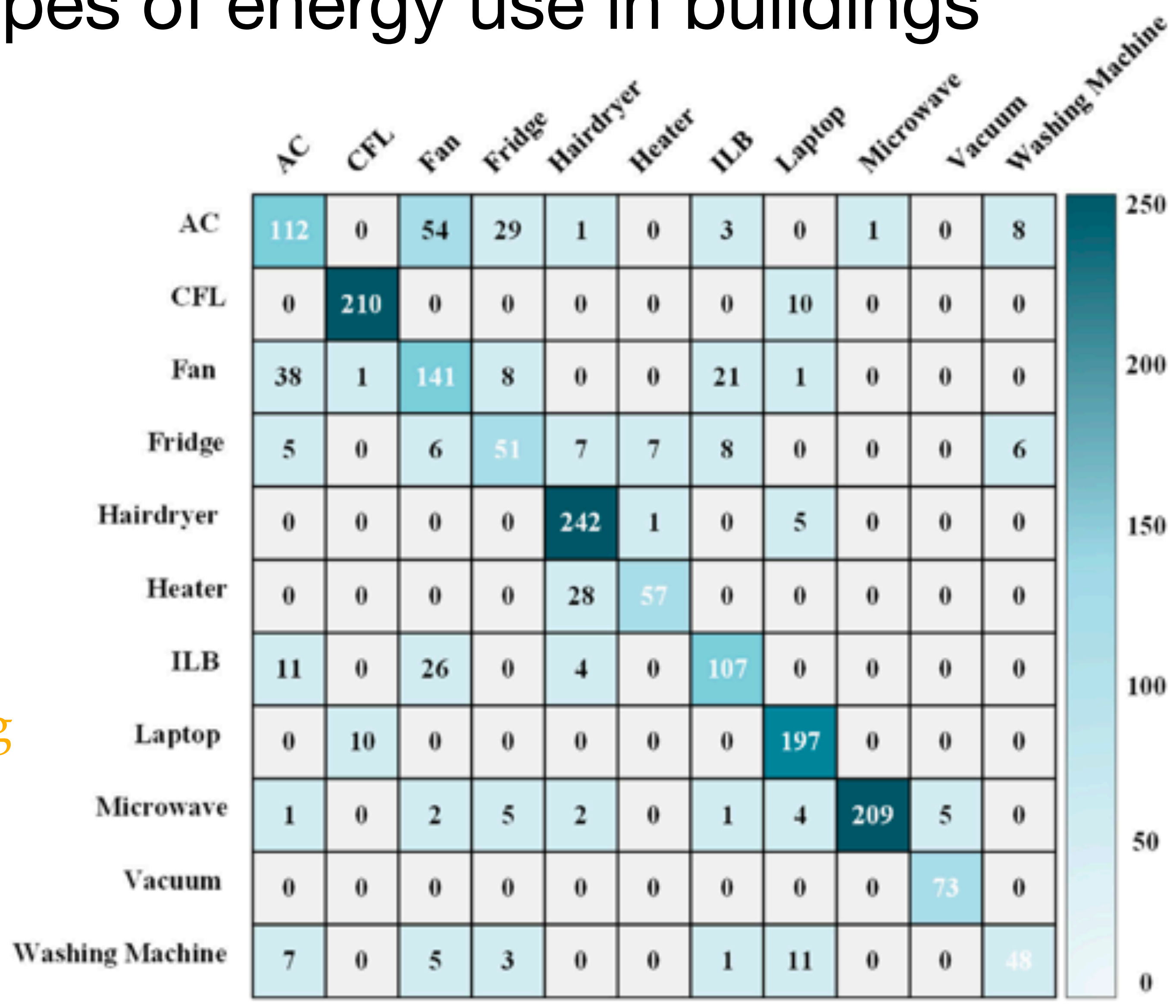
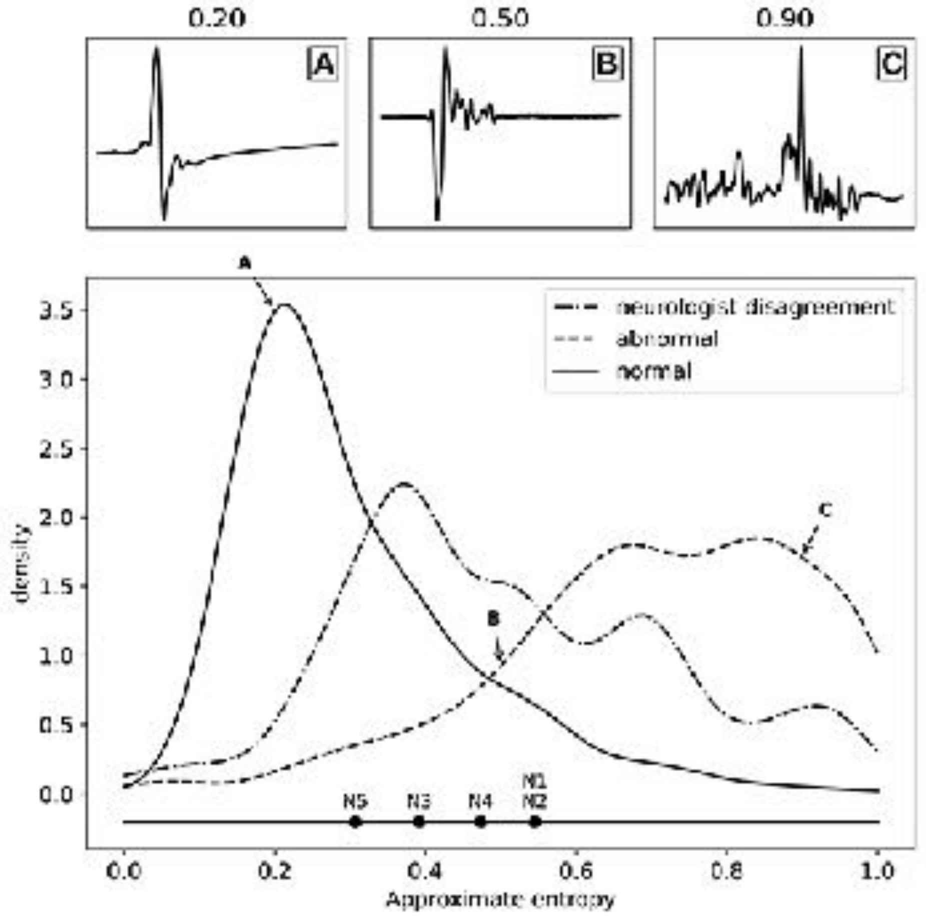


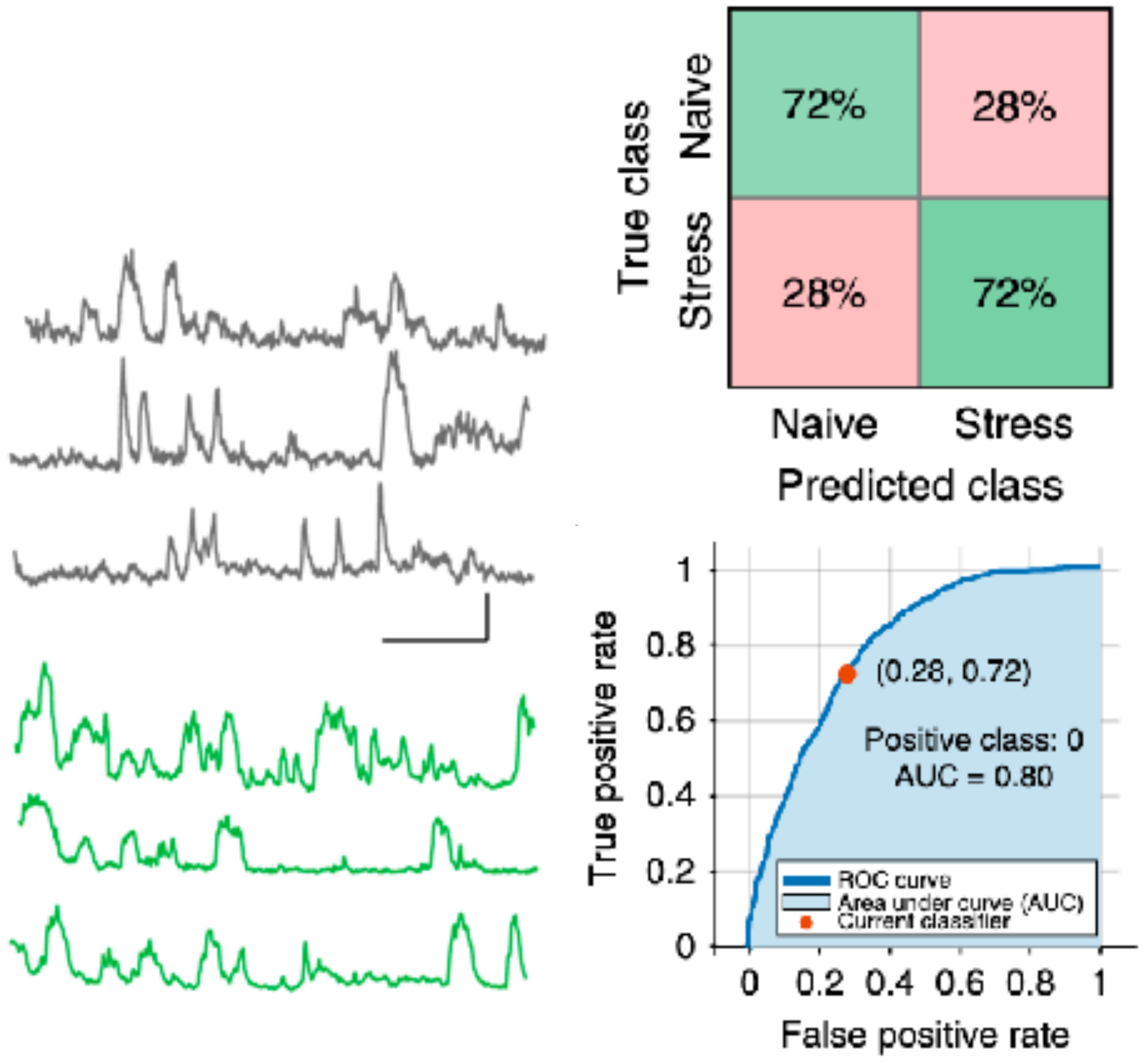
Fig. 7. Confusion matrix of the proposed model.

Distinguish multiple sclerosis MEPs



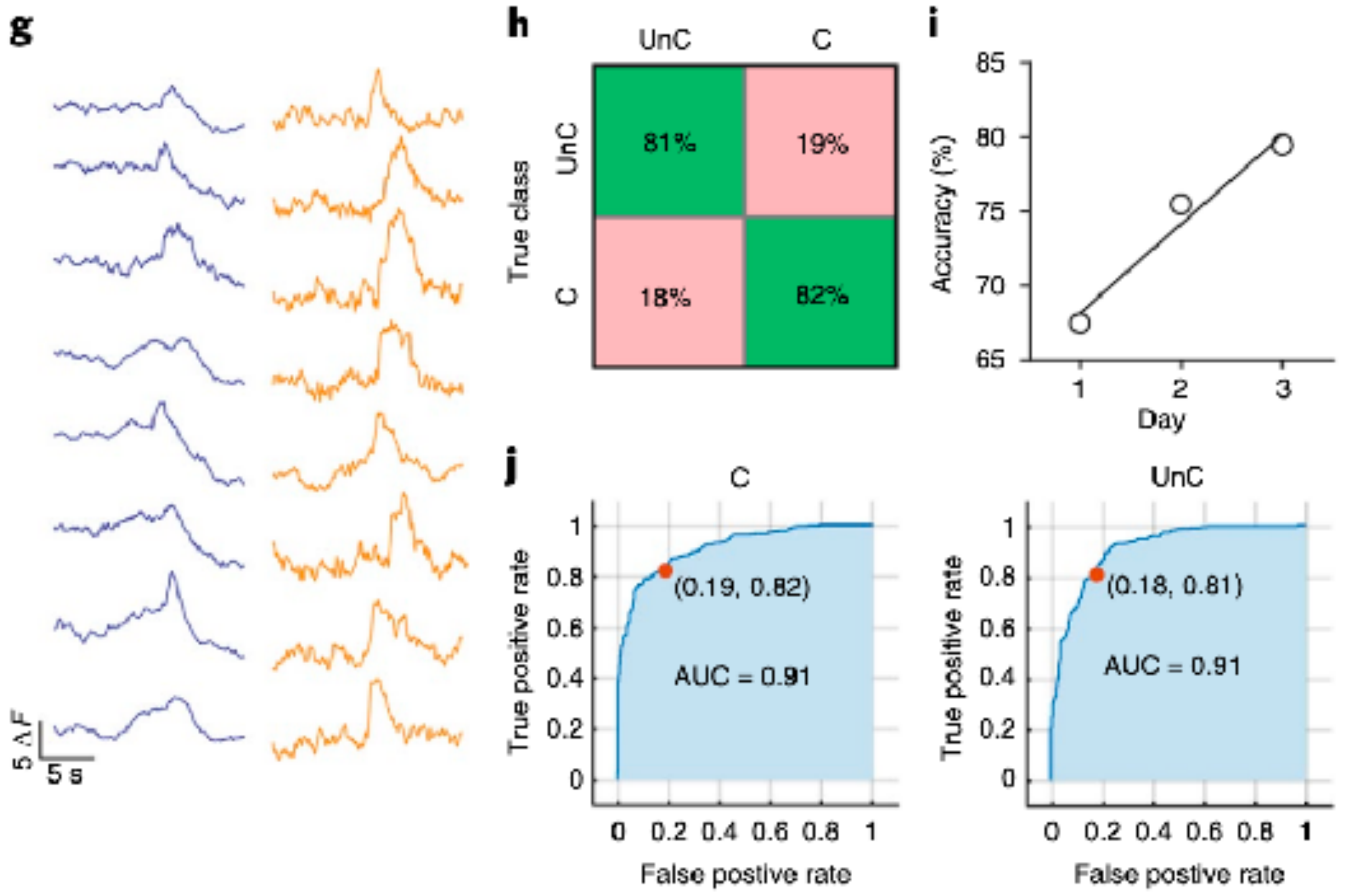
Ypperman et al. Deciphering the Morphology of Motor Evoked Potentials. *Front. Neuroinform.* **14**:28 (2020).

Assess stress-induced changes in astrocyte calcium dynamics



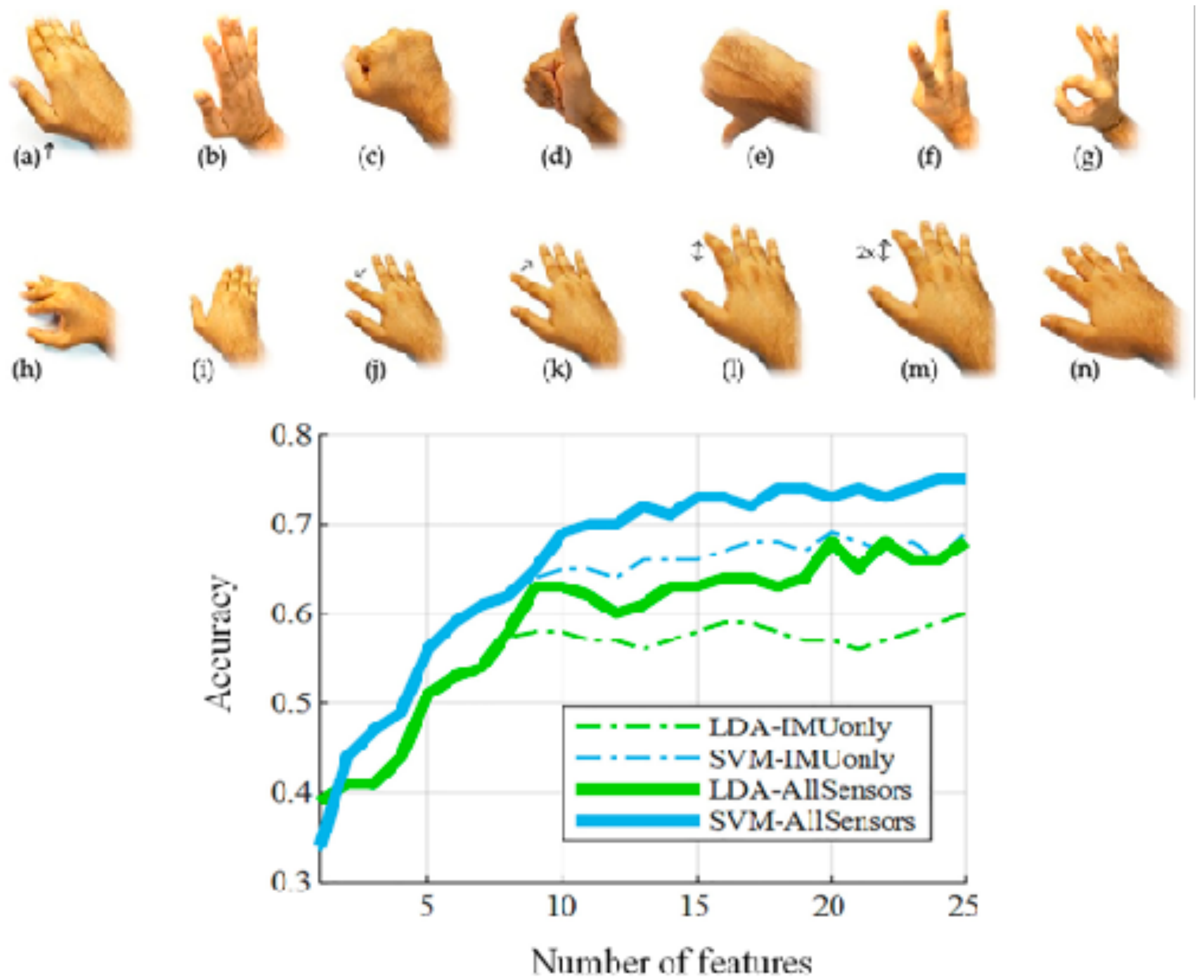
Murphy-Royal et al. Stress gates an astrocytic energy reservoir to impair synaptic plasticity. *Nat Commun* **11**, 2014 (2020).

Assess the stress controllability of neurons



Daviu et al. CRH neurons encode stress controllability and regulate defensive behavior selection. *Nat Neurosci* **23**, 398–410 (2020).

Hand-gesture recognition

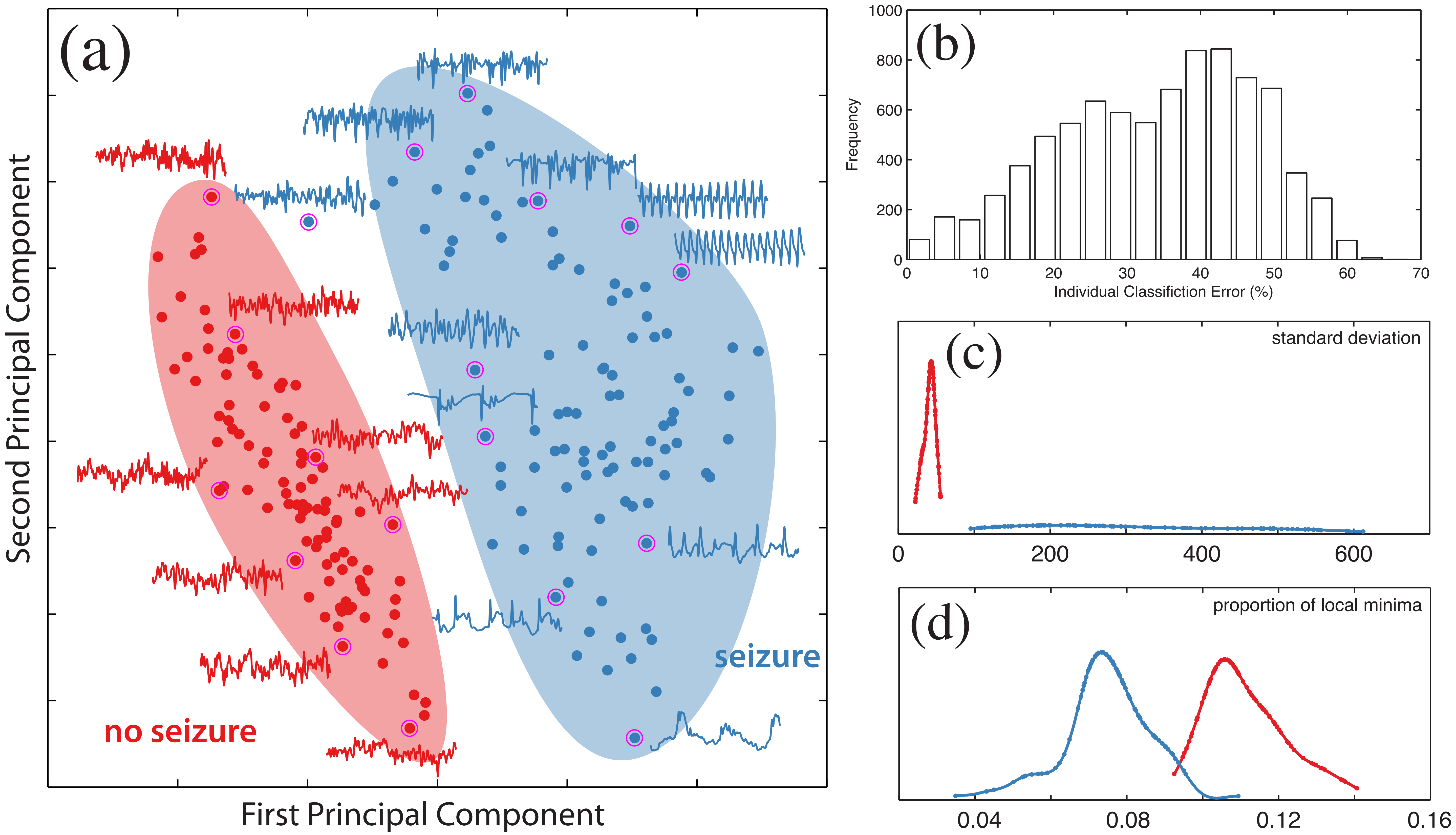


Siddiqui et al. Multimodal hand gesture recognition using single IMU and acoustic measurements at wrist. *PLoS ONE*, **15**, e0227039 (2020).

Case study: Seizure classification

Without comparison, how can I know whether any manually selected analysis method is (close to) optimal?

Without comparison, how can I know whether the complexity in a proposed method is required (over simpler methods)?



discrete wavelet transform features -> ICA, RBF kernel SVM (accuracy > 98%)

"it is likely that methods of this type will be required to configure intelligent devices for treating epilepsy to each individual's neurophysiology prior to clinical operation"

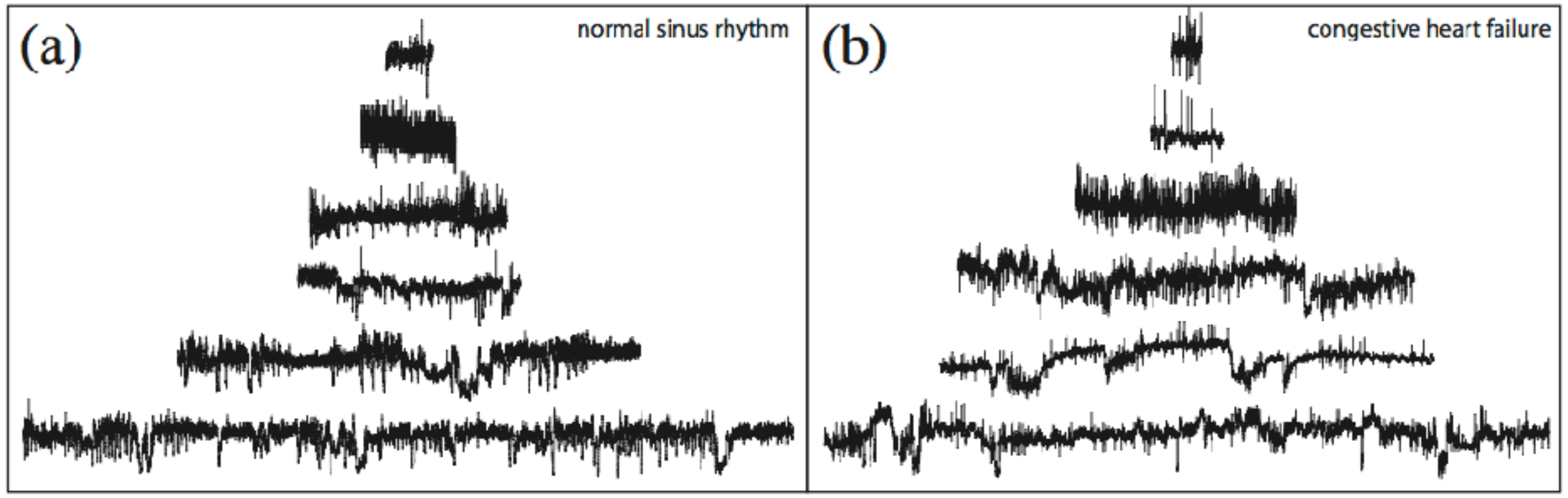
—REDACTED et al. (2010) [>600 citations]

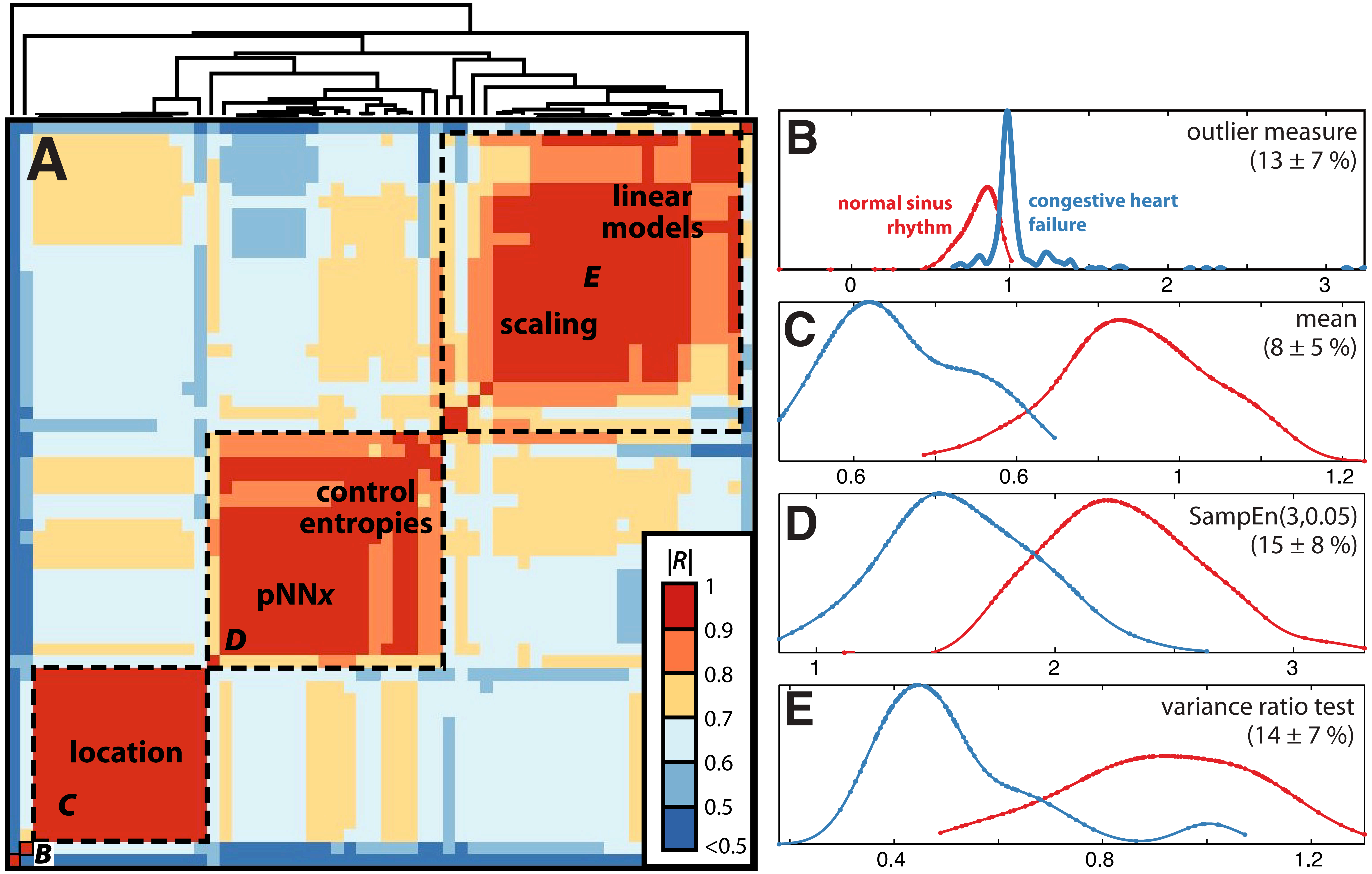
Peer-reviewers and editors should enforce basic levels of comparison when authors make strong claims about the relative utility of their methods

Case study:
Heart rate variability

Without comparison, how can I know whether methods with different-sounding names are actually unique, or whether new methods actually reproducing the behavior/performance of existing methods?

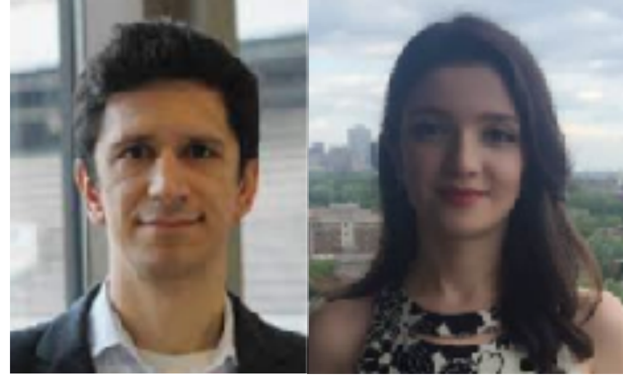
Instead of many papers on a topic, you can write a single paper! 🥳

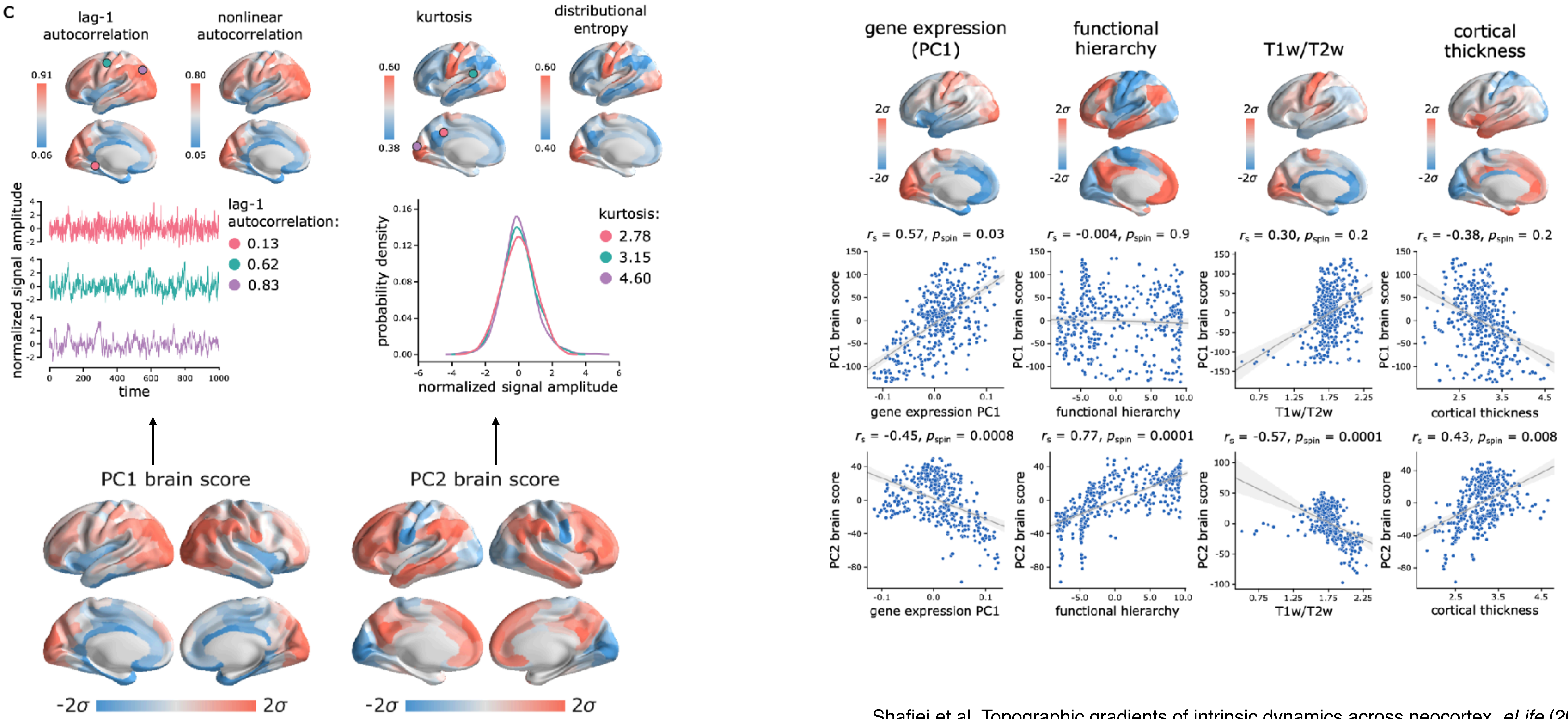




Low-dimensional feature-space projections

Infer shared variance across diverse time-series properties

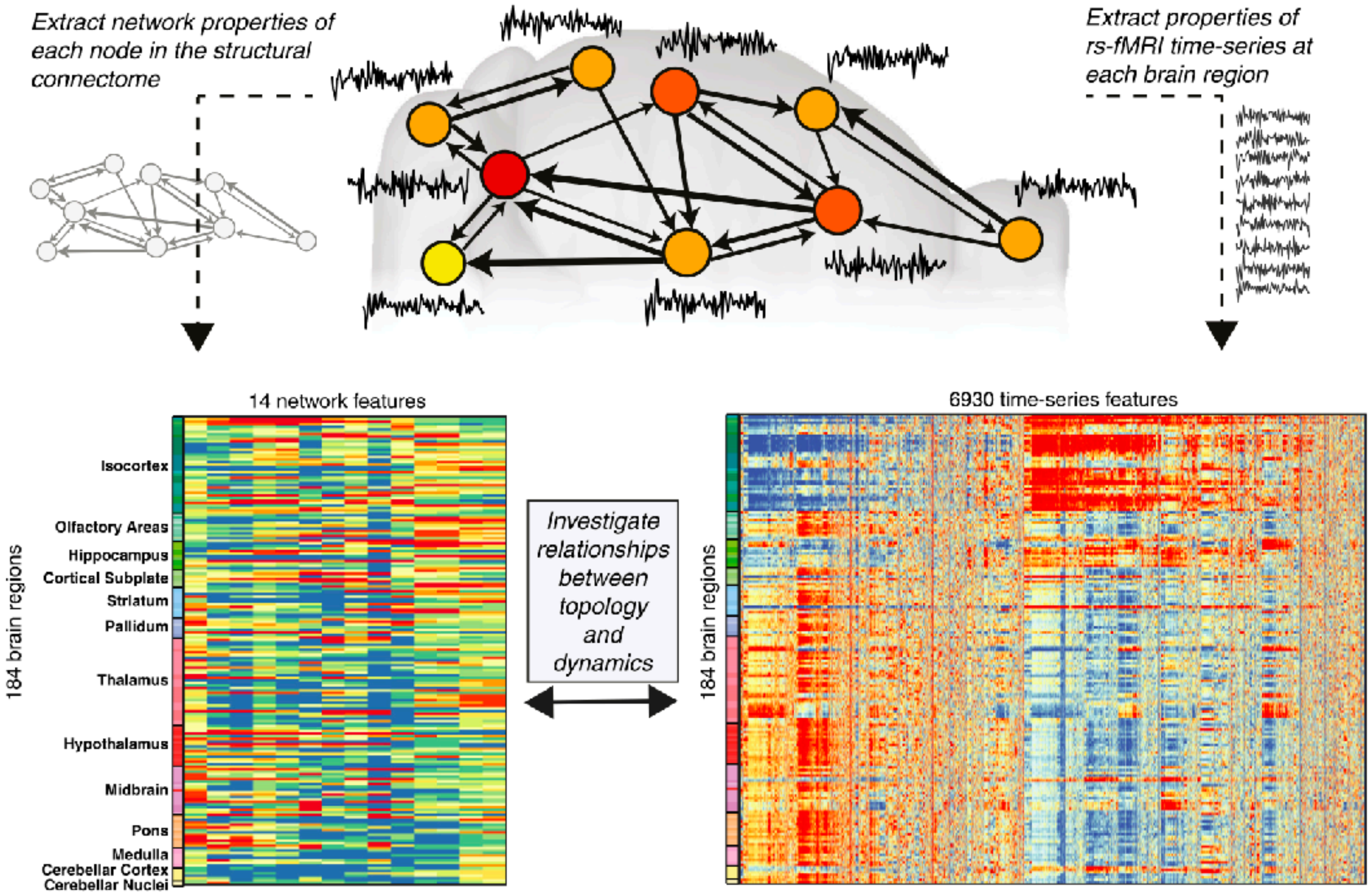


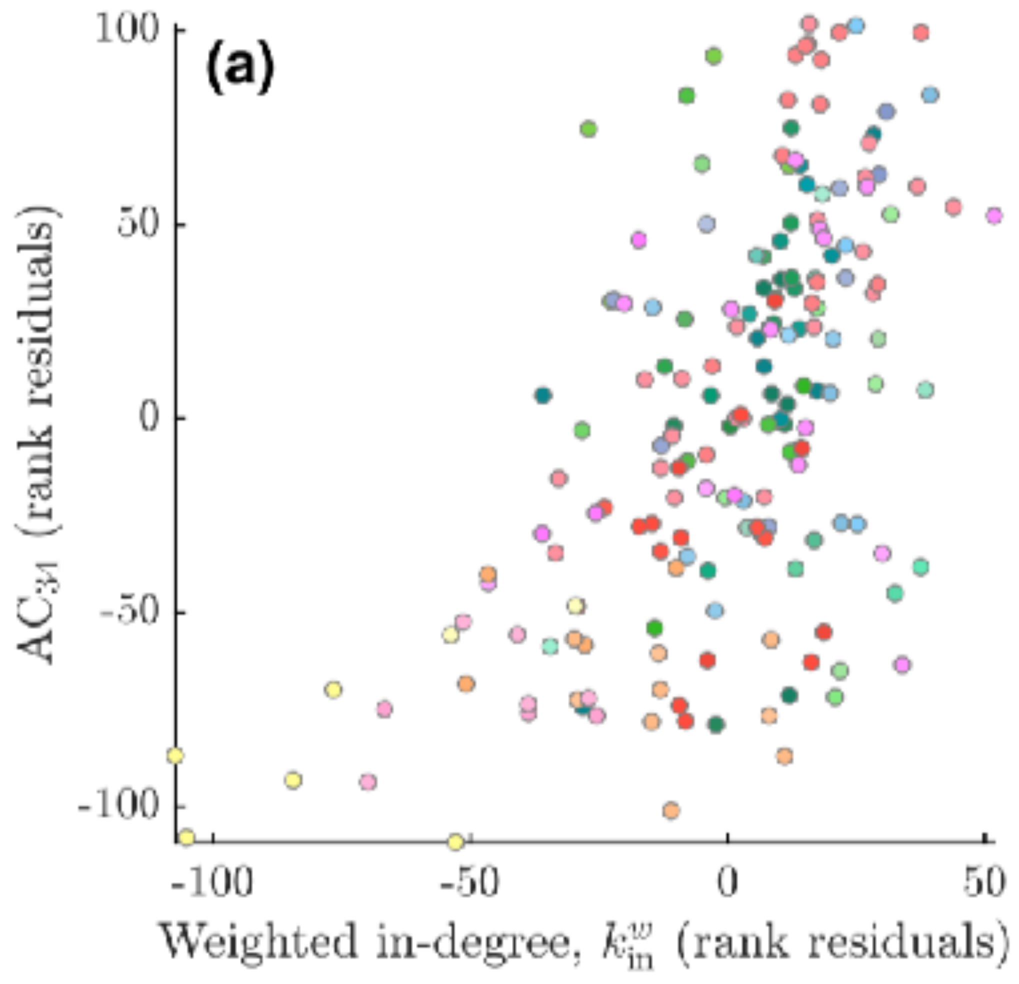


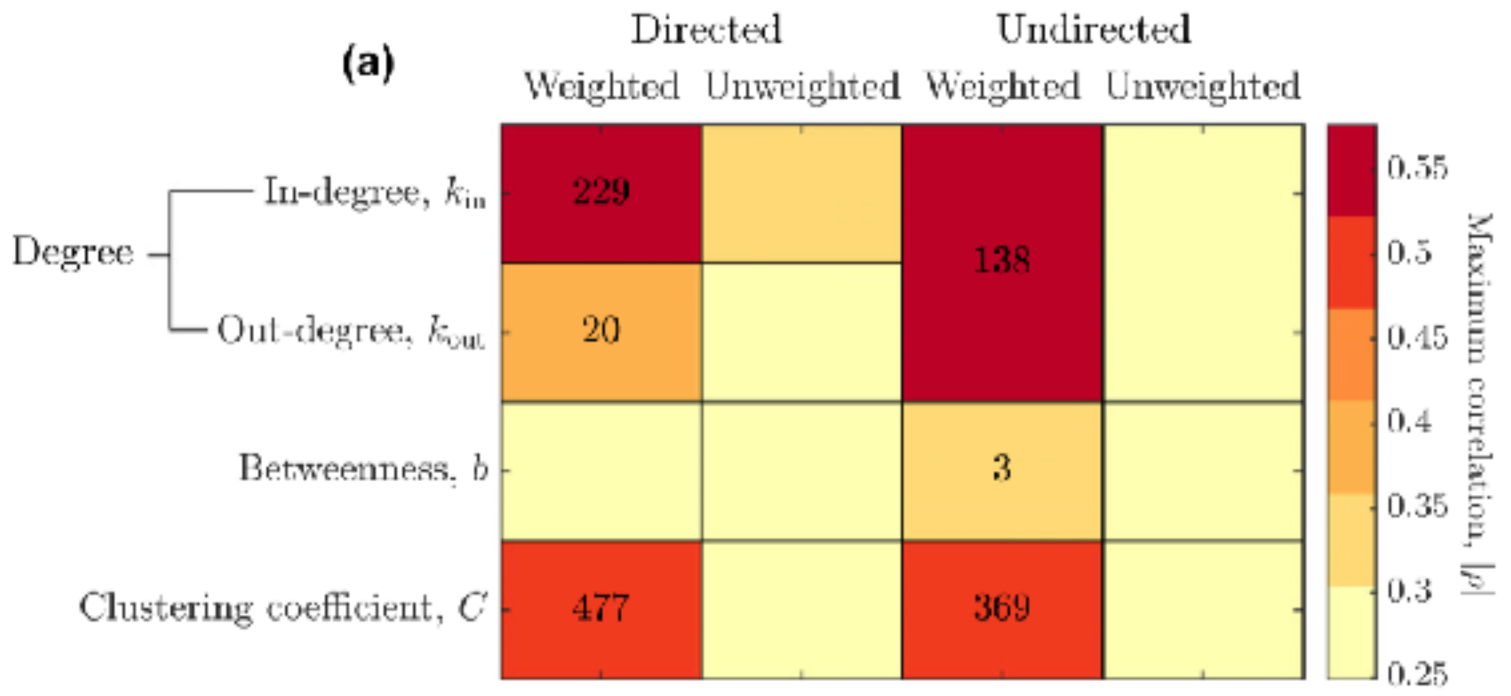
Shafiei et al. Topographic gradients of intrinsic dynamics across neocortex. *eLife* (2020)

fMRI signatures of brain connectivity: mouse

What types of structural connectome properties correlate with what types of local time-series dynamics?



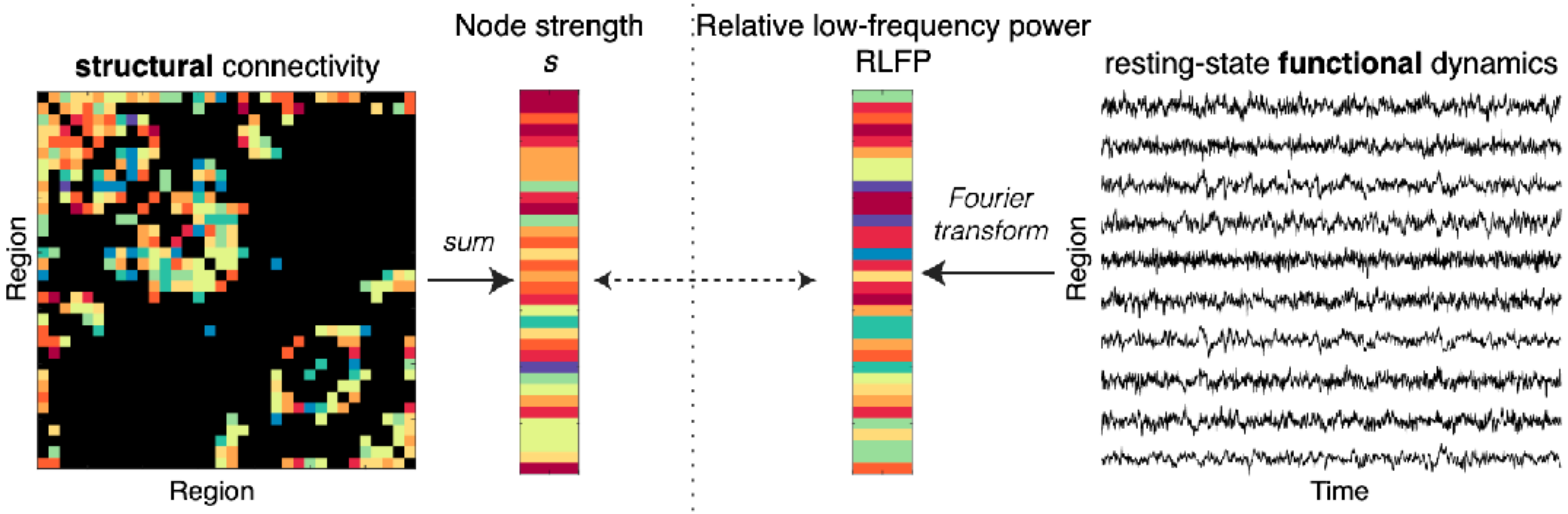


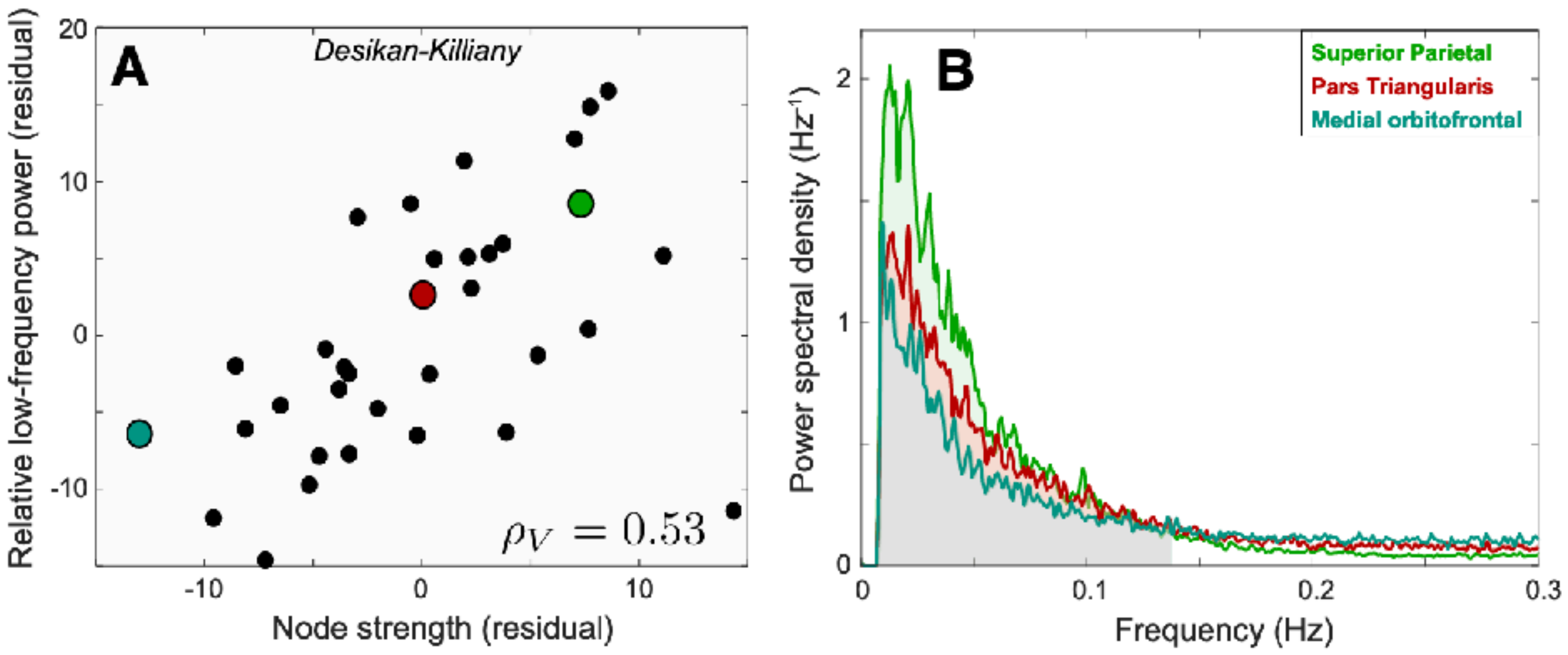


Sethi et al. (2017). Structural connectome topology relates to regional BOLD signal dynamics in the mouse brain. *Chaos.*

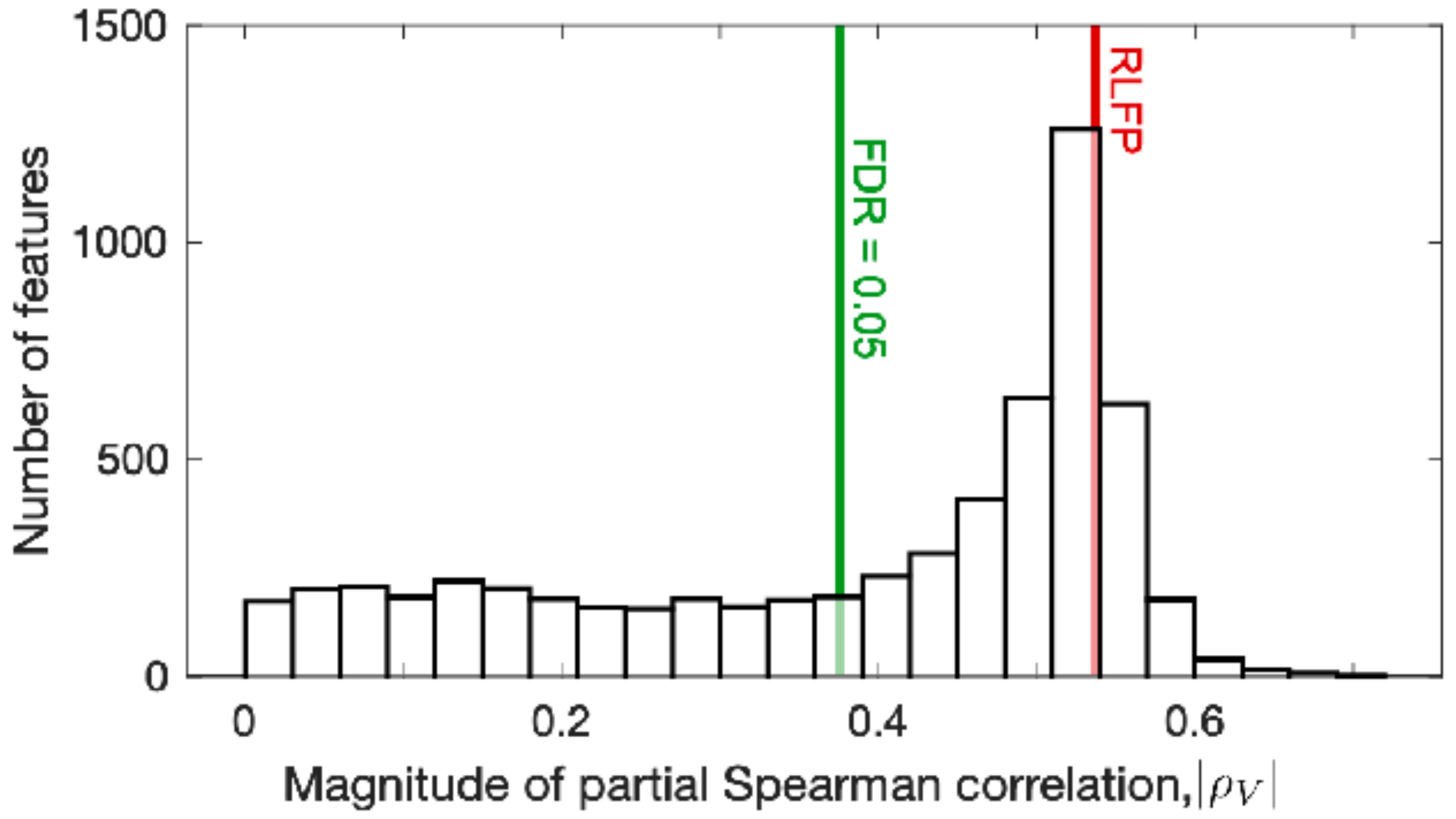
fMRI signatures of brain connectivity: human

Do we see similar types of relationships between local dynamics and local connectome properties in human cortex?



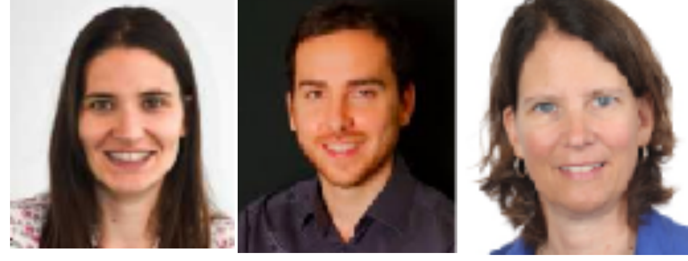


How good is the target feature relative to alternatives?

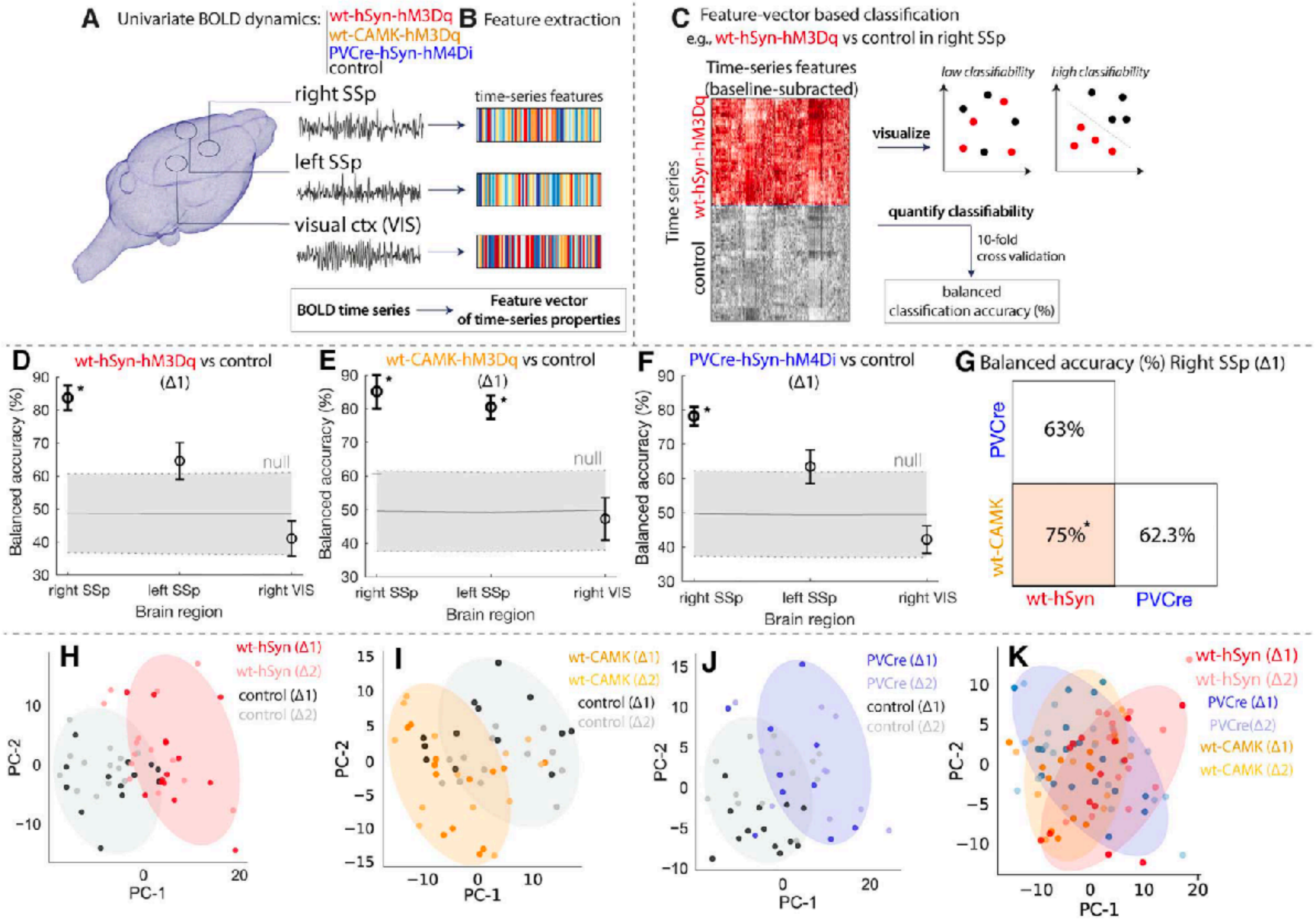


Fallon et al. (2020). Timescales of spontaneous fMRI fluctuations relate to structural connectivity in the brain. *Net. Neurosci.*

BOLD signatures of E:I balance



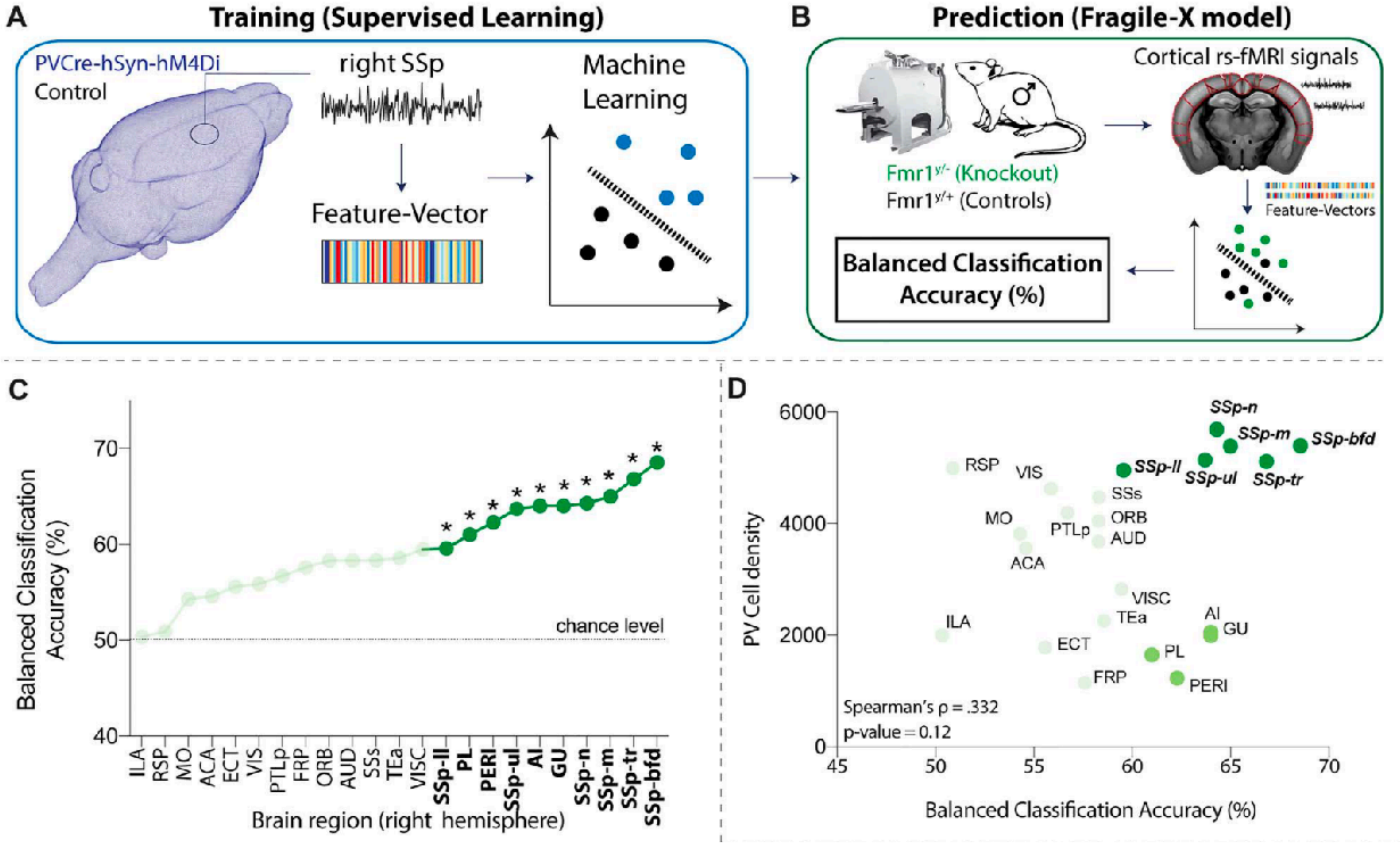
Extracting cellular-level information (E:I balance from chemogenetic manipulations) from macroscopic BOLD dynamics



Markicevic et al. (2020). Cortical Excitation:Inhibition imbalance causes abnormal brain network dynamics as observed in neurodevelopmental disorders. *Cerebral Cortex*.

BOLD signatures of E:I balance

Learned classification rules from targeted chemogenetic manipulations accurately distinguish fMRI dynamics in a gene knockout mouse

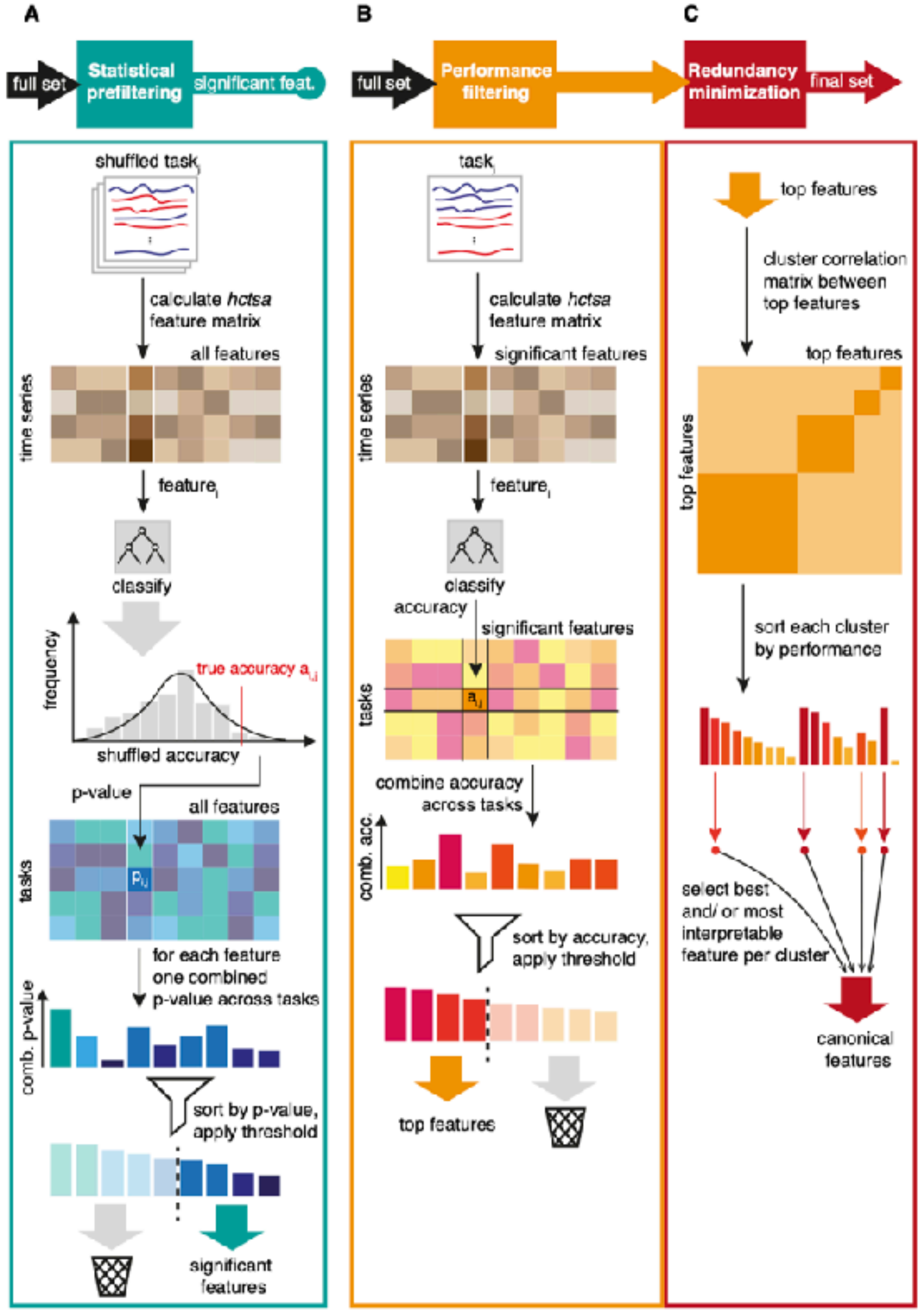


Markicevic et al. (2020). Cortical Excitation:Inhibition imbalance causes abnormal brain network dynamics as observed in neurodevelopmental disorders. *Cerebral Cortex*.

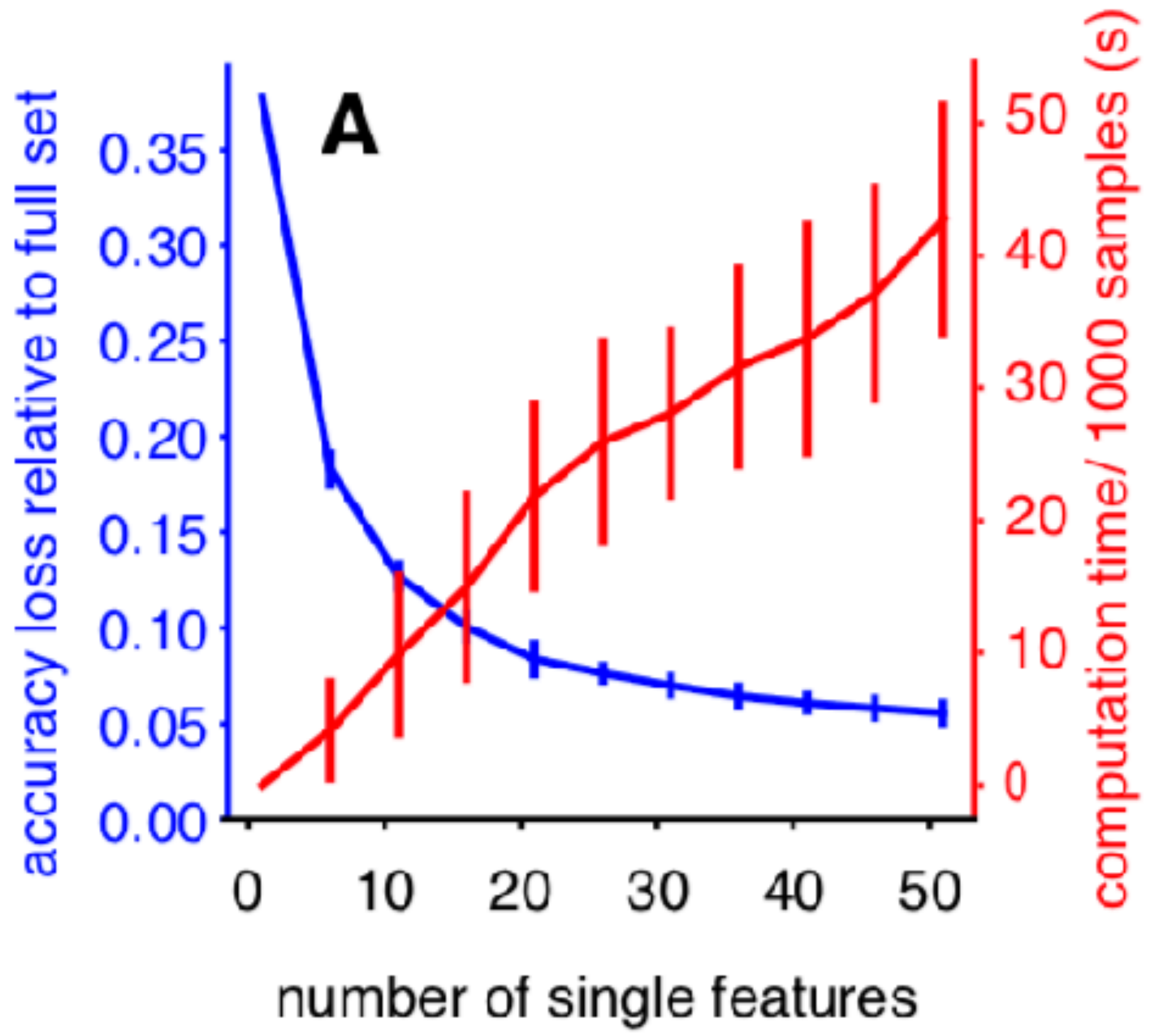
>7000 Features?

Is 7000 features too many?

Yes.



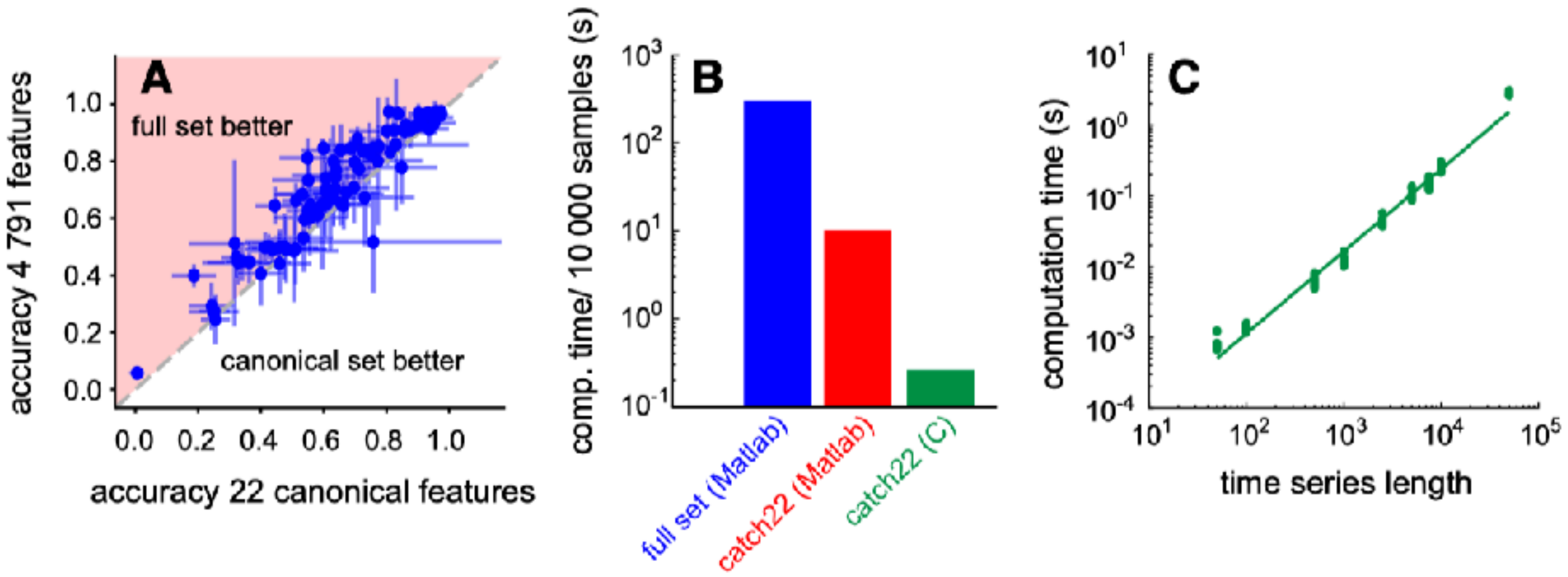
Feature evaluation across 93 time-series classification tasks



Lubba, C. H. et al. *catch22*: CAnonical Time-series CHaracteristics. *Data Min Knowl Disc* **33**, 1821–1852 (2019).

catch22

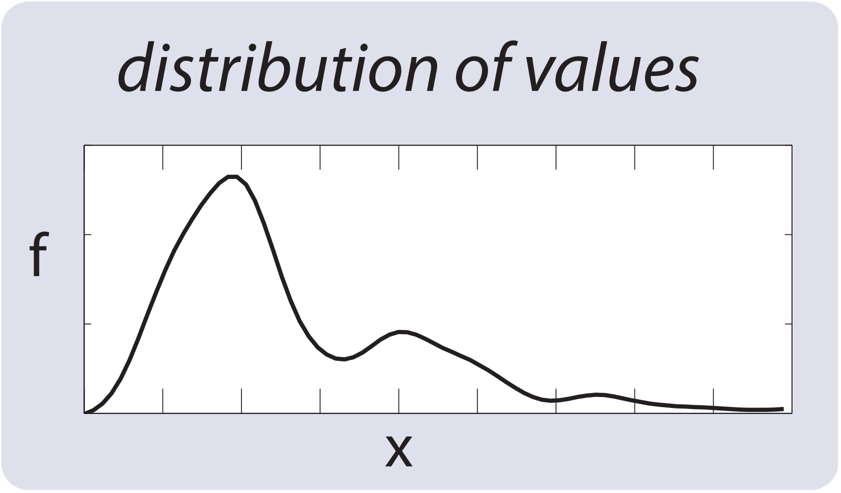
We mostly get away with just 22 features.
C-coded for efficiency.



IMPORTANT: location/spread-dependent features are not included

When these (non-dynamical) properties are important for classification, catch22 will appear very poor: should add them (now easy as **switching on catch24**)

```
catch22_all(data, catch24=True)
```



Distribution

DN_HistogramMode_5

DN_OutlierInclude_p_001_mdrmd

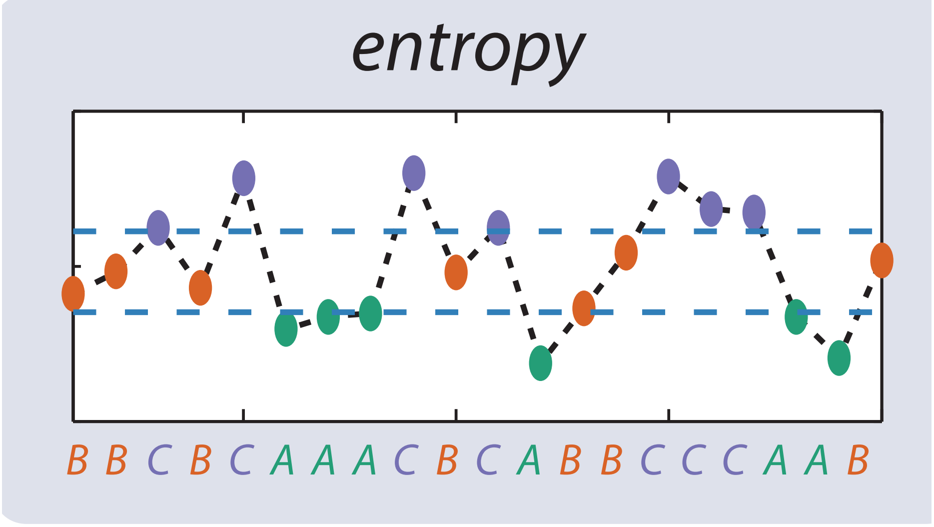
DN_HistogramMode_10

DN_OutlierInclude_n_001_mdrmd

Fluctuation Analysis

SC_FluctAnal_2_dfa_
50_1_2_logi_prop_r1

SC_FluctAnal_2_rsrangefit_
50_1_logi_prop_r1



Temporal Statistics

SB_BinaryStats_mean_longstretch1

SB_TransitionMatrix_3ac_sumdiagcov

PD_PeriodicityWang_th0_01

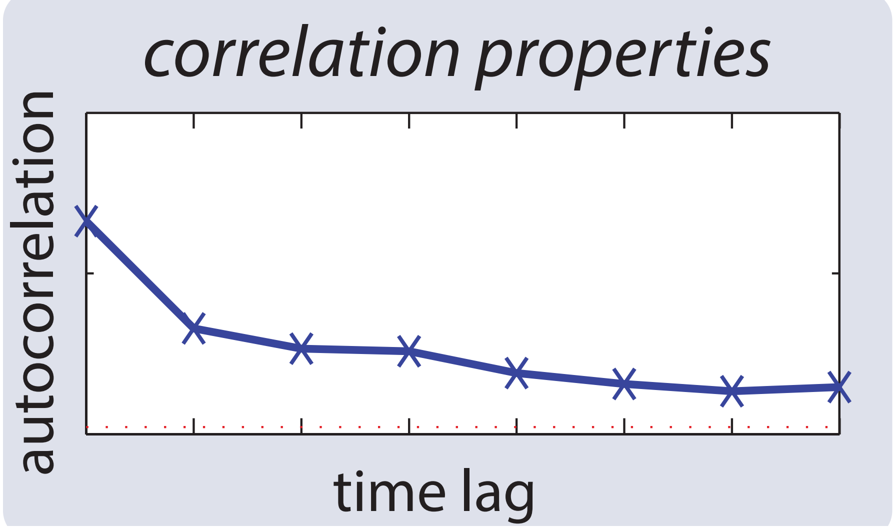
MD_hrv_classic_pnn40

SB_BinaryStats_diff_longstretch0

SB_MotifThree_quantile_hh

FC_LocalSimple_mean1_tauresrat

CO_Embed2_Dist_tau_d_expfit_meandiff



Linear Autocorrelation

CO_f1ecac

CO_FirstMin_ac

SP_Summaries_welch_rect_area_5_1

SP_Summaries_welch_rect_centroid

FC_LocalSimple_mean3_stderr

Nonlinear Autocorrelation

CO_trev_1_num

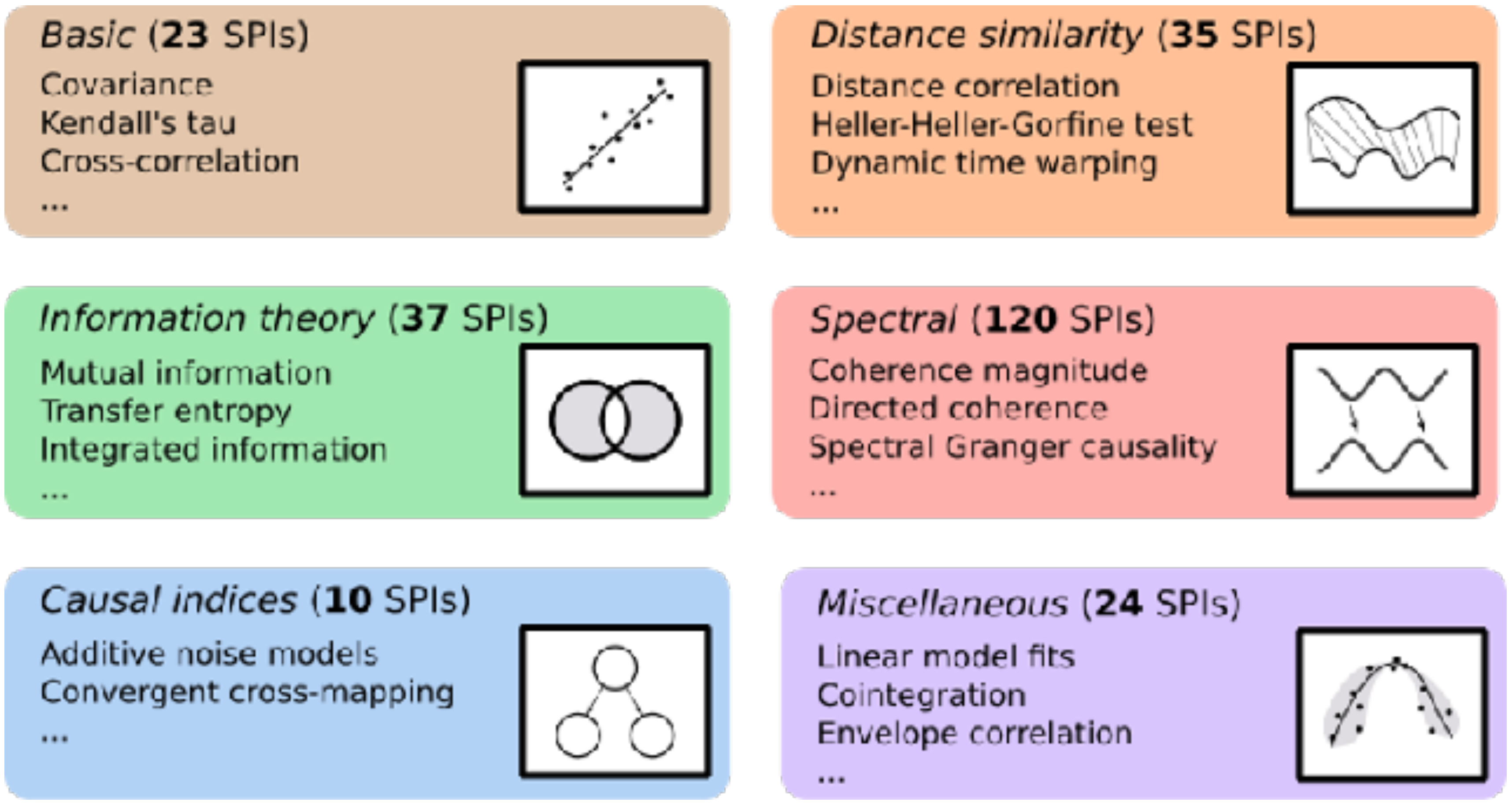
CO_HistogramAMI_even_2_5

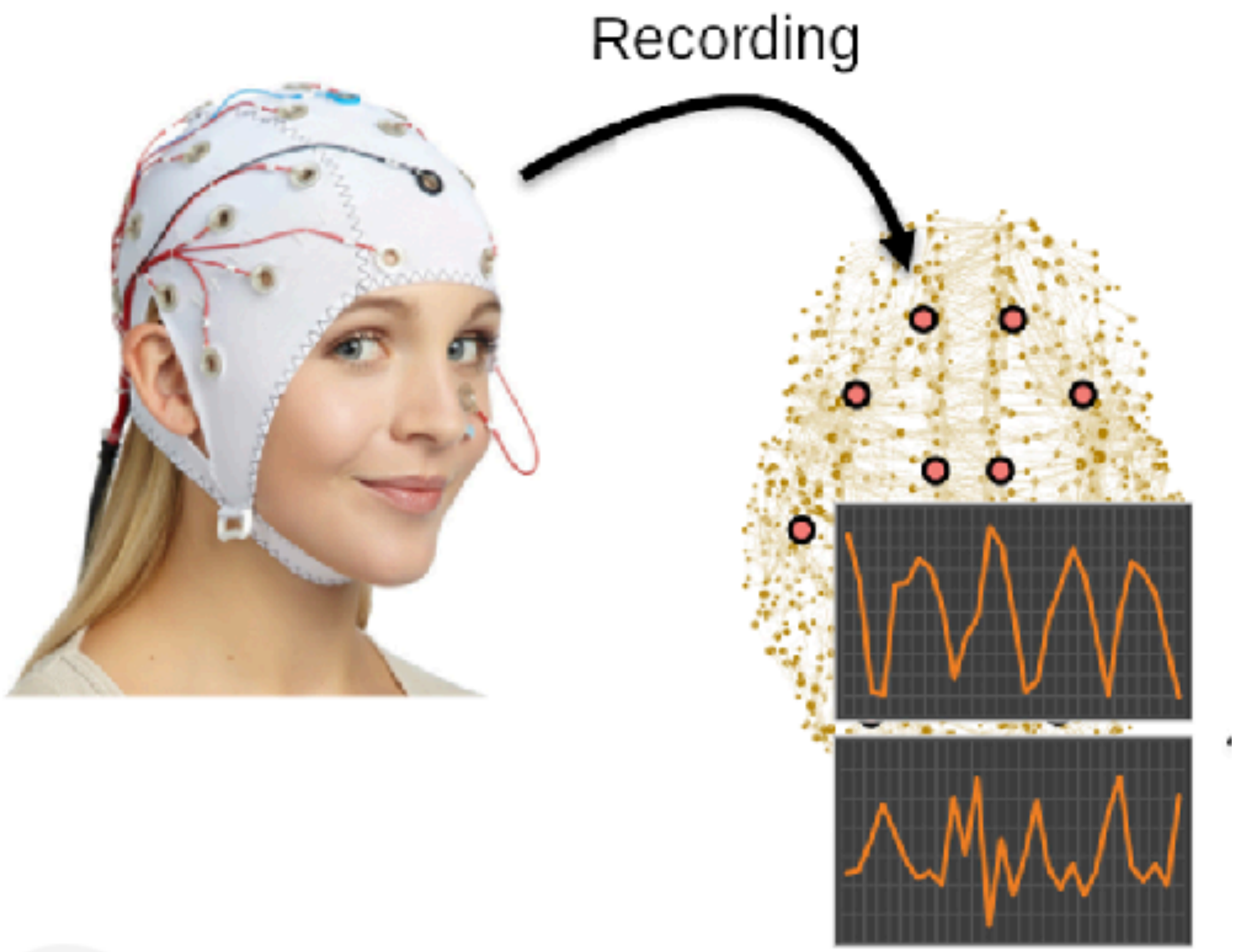
IN_AutoMutualInfoStats_
40_gaussian_fmmi

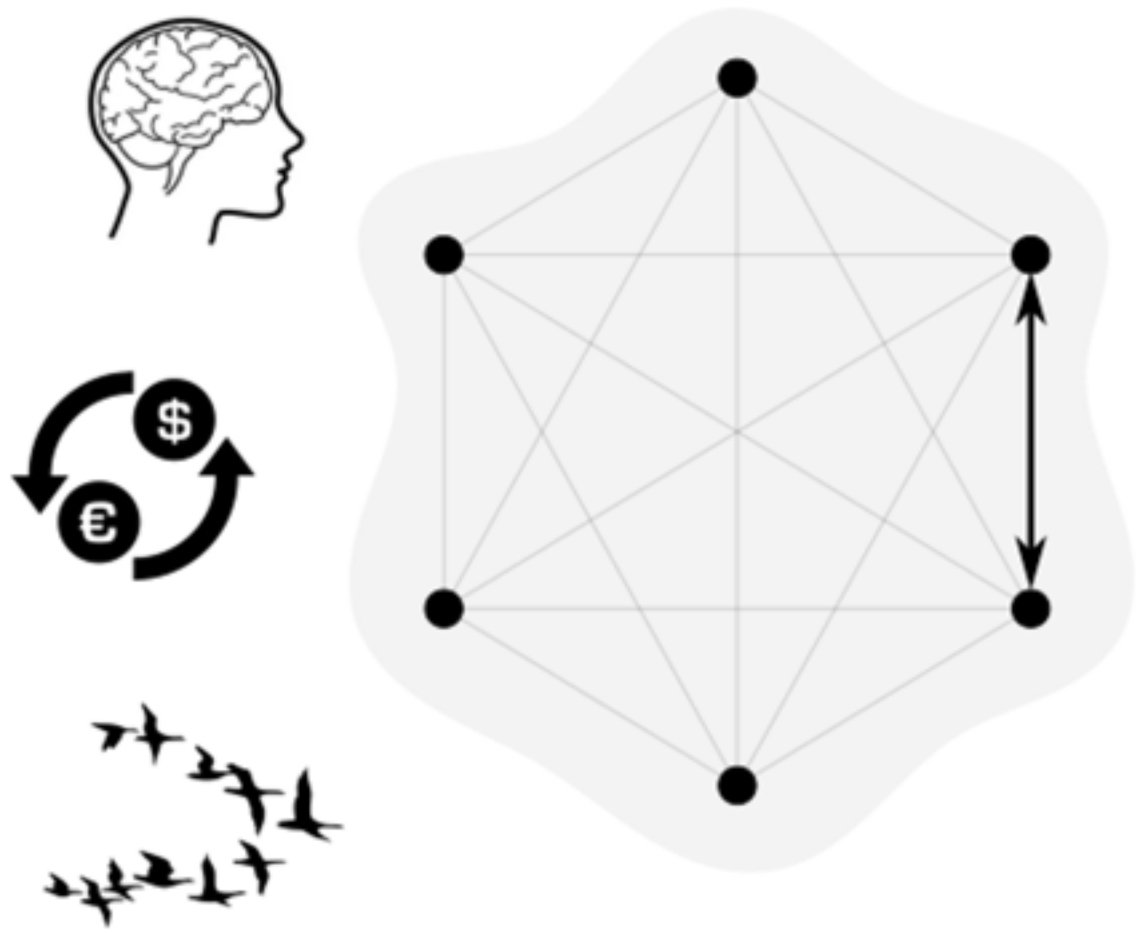
Lubba, C. H. et al. *catch22*: CAnonical Time-series CHaracteristics. *Data Min Knowl Disc* **33**, 1821–1852 (2019).

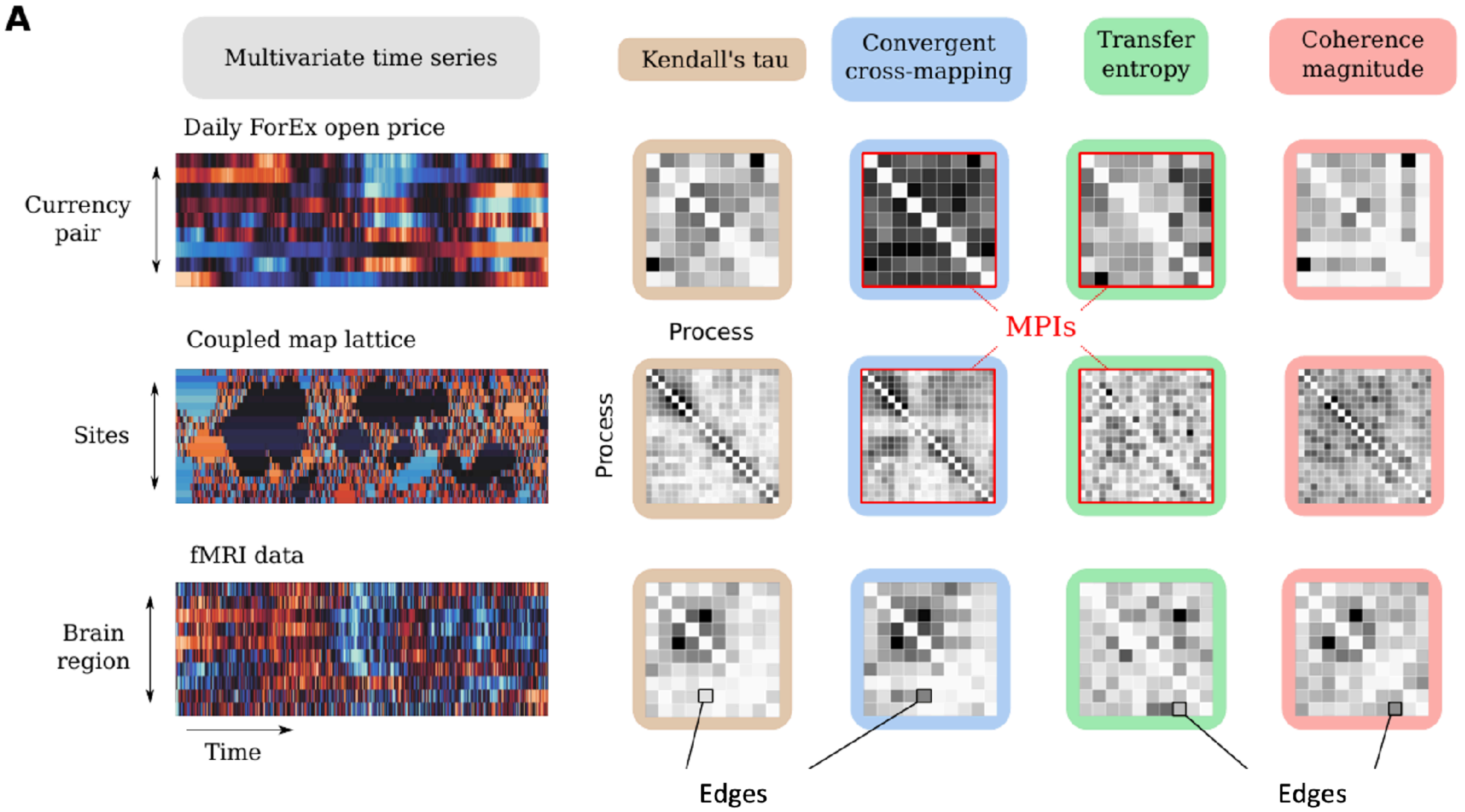
What about for multi-component systems?

- Prior work considered a single dynamical process in isolation (yielding **univariate** time series).
- But real systems involve multiple interacting processes, measured as **multivariate** time series.
- Many methods exist to capture pairwise relationships between elements of a system.
    - Most common being a Pearson cross-correlation.
- We collected a library of ~250 other measures.








A
Multivariate time series
Kendall's tau
Convergent cross-mapping
Transfer entropy
Coherence magnitude
Daily ForEx open price
Currency pair
Coupled map lattice
Sites
fMRI data
Brain region
Time
Process
Process
MPIs
Edges
Edges

pyspi

Beyond univariate features—how do elements of the system interact?

Python Toolkit of Statistics for Pairwise Interactions (pyspi)

DOI 10.5281/zenodo.5787486

PySPI is a comprehensive library for computing pairwise interactions from multivariate time-series (MTS) data.

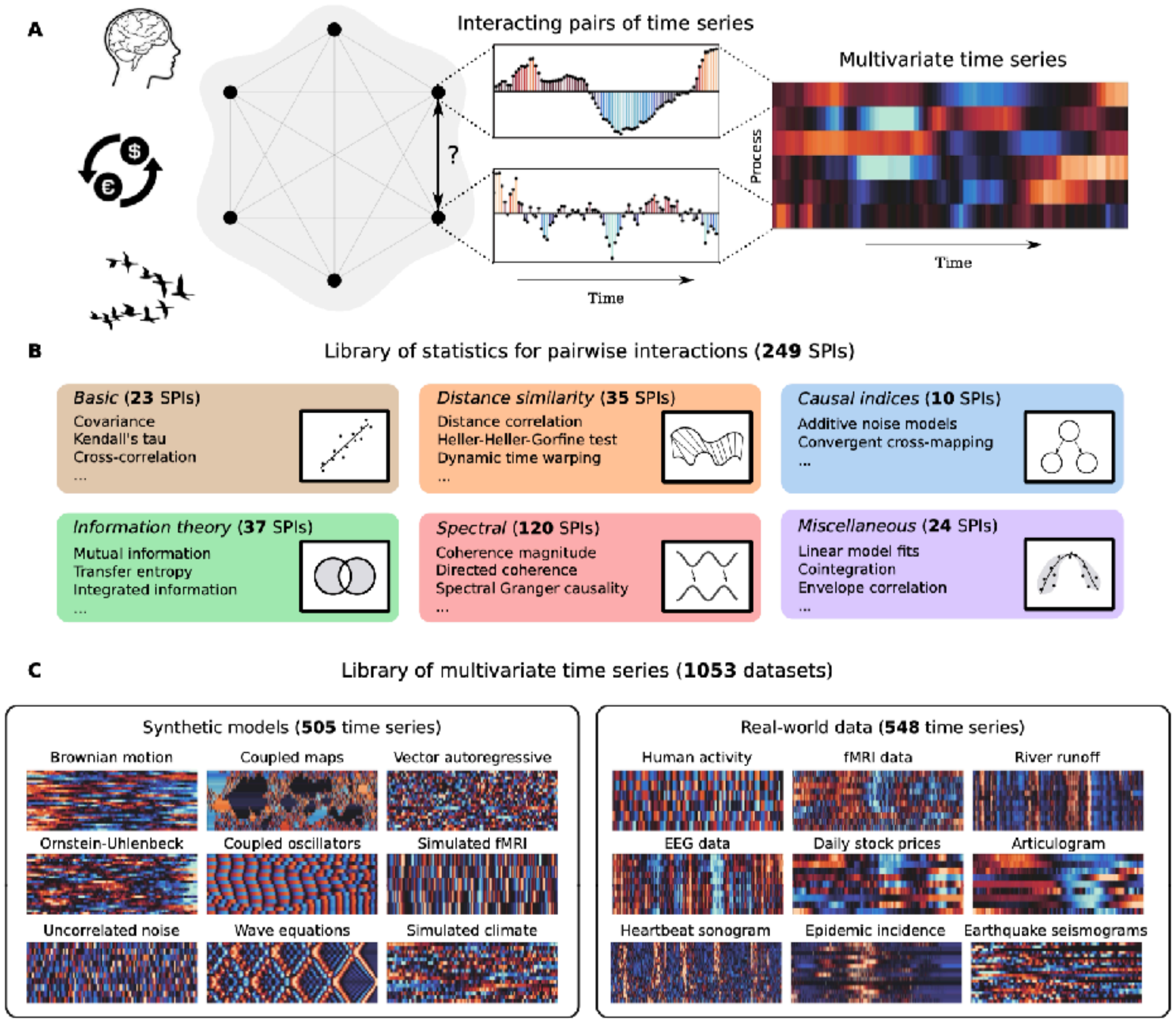
The code provides easy access to hundreds of methods for evaluating the relationship between pairs of time series, from simple statistics (like correlation) to advanced multi-step algorithms (like Granger causality). The code is licensed under the GNU GPL v3 license (or later).

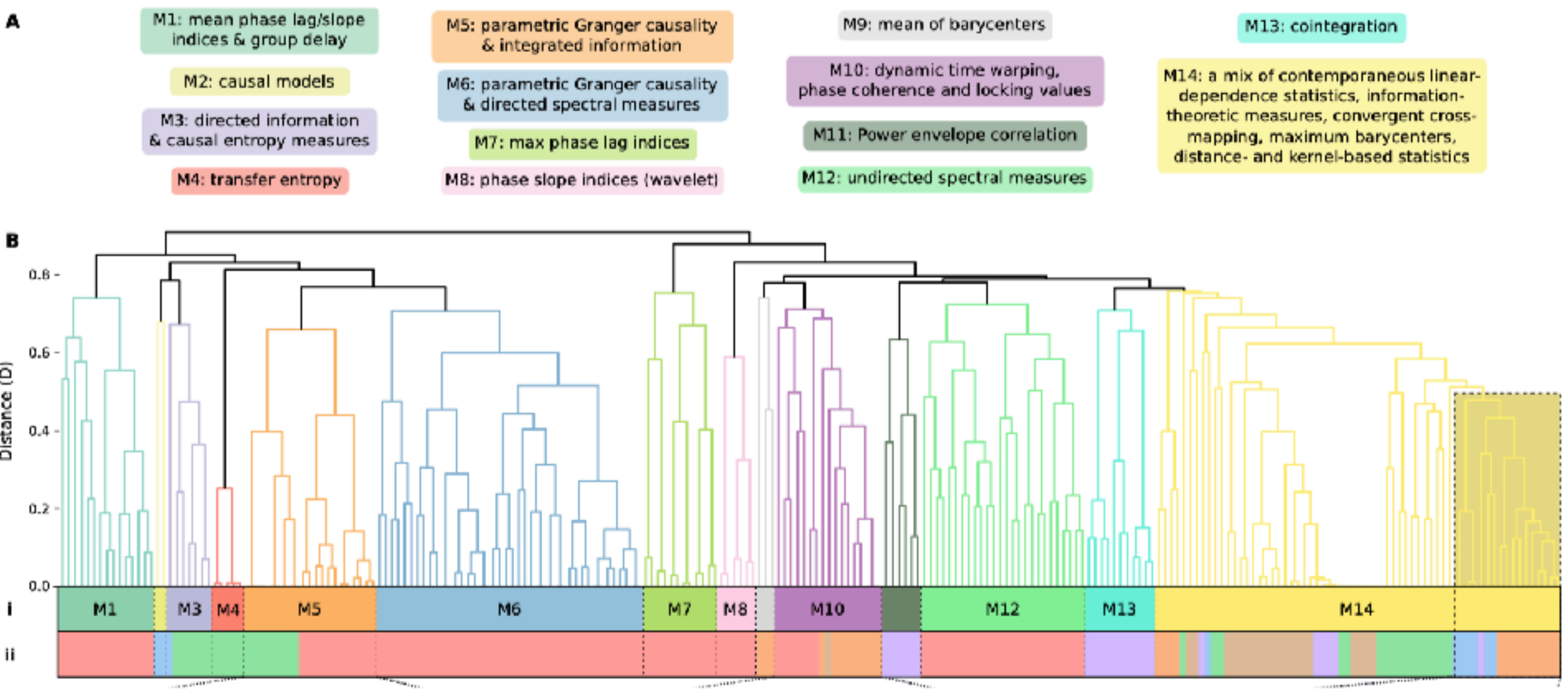
Feel free to email me **for help with real-world applications.** Feedback is much appreciated through email, issues, or pull requests.

Acknowledgement

If you use this code, please cite the following preprint:

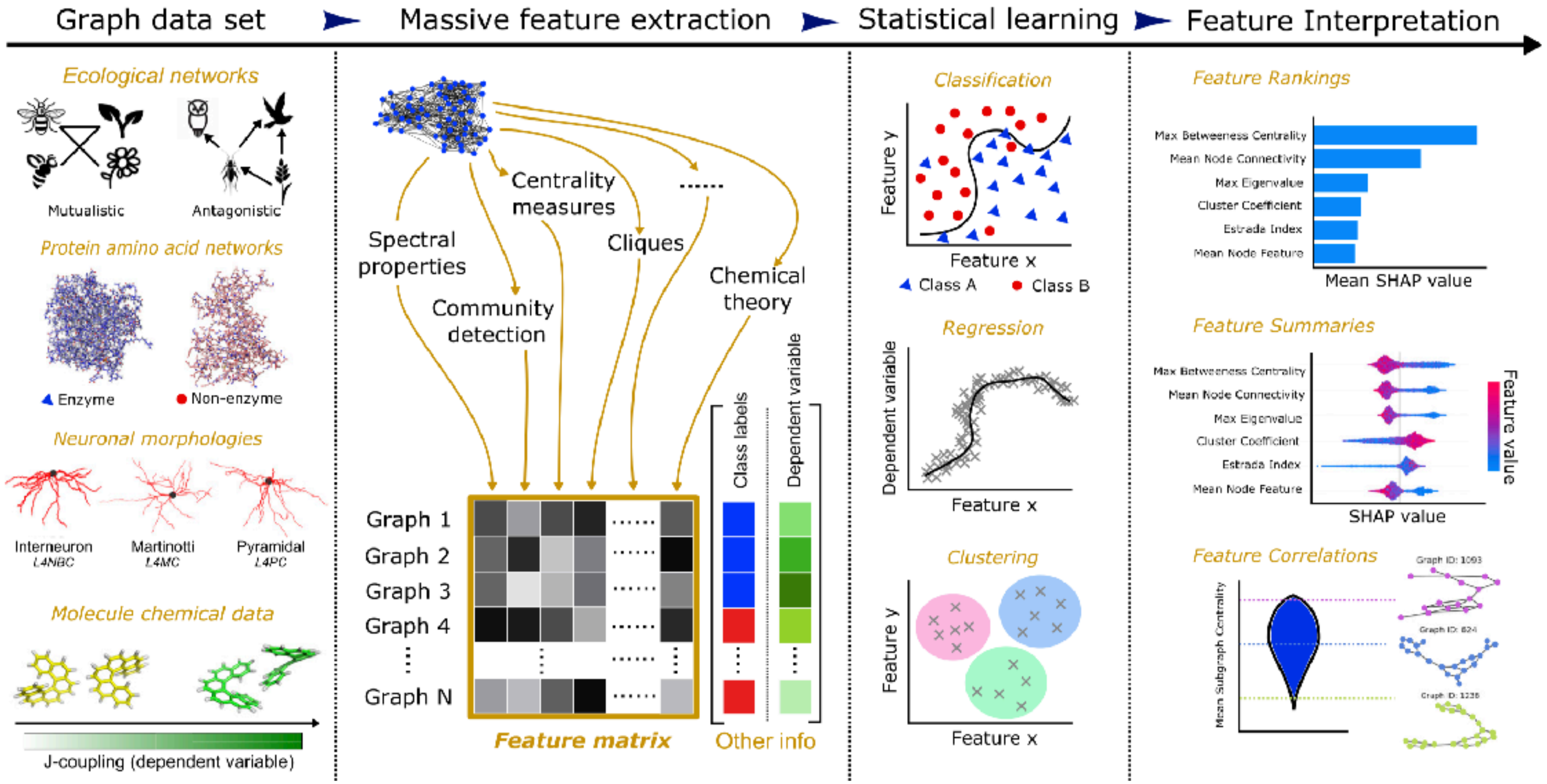
Oliver M. Cliff, Joseph T. Lizier, Naotsugu Tsuchiya, Ben D Fulcher, "Unifying Pairwise Interactions in Complex Dynamics," ArXiv preprint, arXiv:2201.11941 (2022).





What about graphs?

- There are also hundreds of measures to capture structure in graphs (networks)
- Shout out to **hcga:** highly comparative graph analysis for network phenotyping.



Peach et al. (2021). HCGA: Highly comparative graph analysis for network phenotyping. *Patterns*.

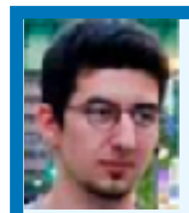
Our Feature-Based Time-Series Analysis Tools

hctsa

Compute >**7700** time-series features

Low-dimensional projections
Classification, …

https://github.com/benfulcher/hctsa



catch22

Compute **22** time-series features

Fast-coded in C

+ mean & std (as *catch24)*

pycatch22

```
pip install pycatch22
```

Rcatch22

```
install.packages("Rcatch22")
```

Catch22.jl

```
Pkg.add("Catch22")
```

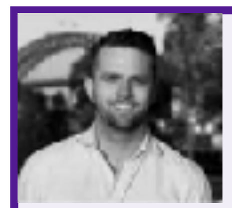
https://github.com/DynamicsAndNeuralSystems/catch22

theft

Feature computation, analysis, and visualization for feature-based time-series analysis.

Includes a range of feature sets: tsfeatures, feasts, tsfresh, TSFEL, catch22, & Kats.

https://github.com/hendersontrent/theft

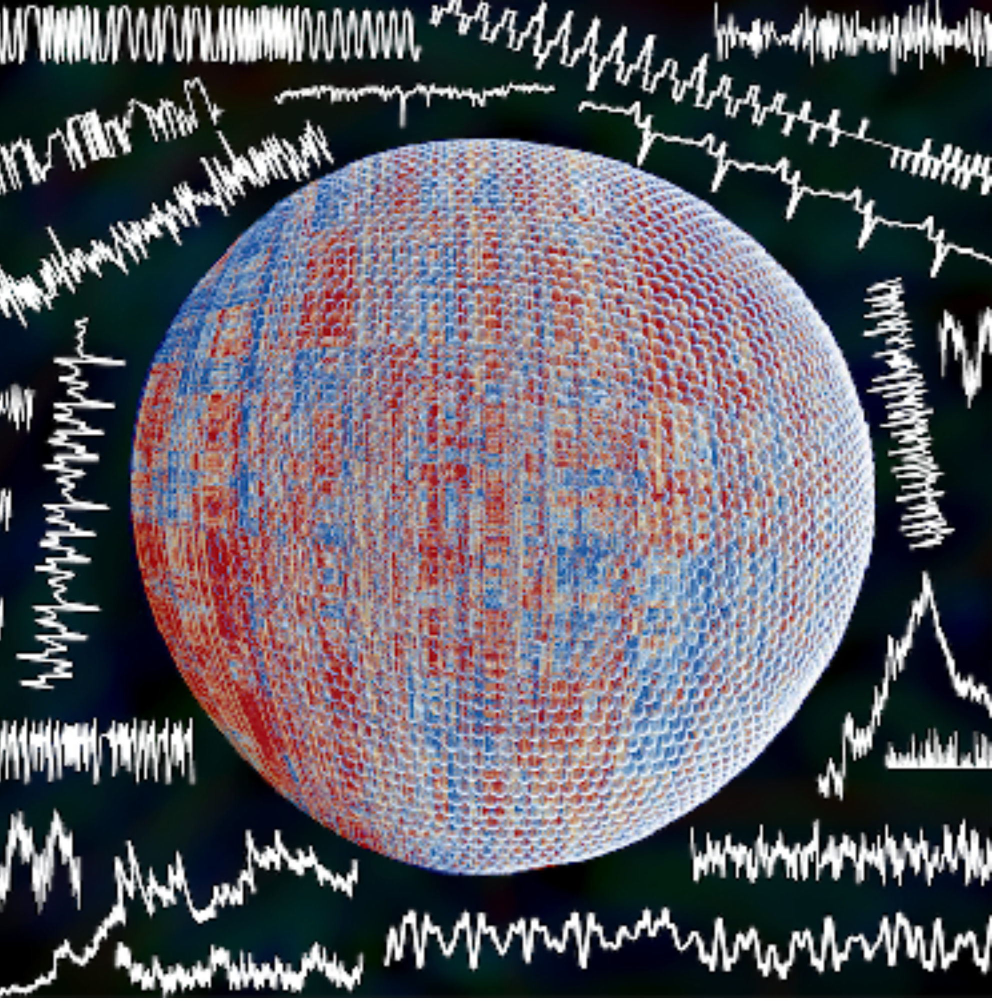


pyspi

Extension to multivariate time series: measures of **pairwise interactions** in a complex system

https://github.com/olivercliff/pyspi

Others have tools in this space also: https://github.com/benfulcher/hctsa/wiki/Related-time-series-resources



Acknowledgements

ben.fulcher@sydney.edu.au
@bendfulcher
@compTimeSeries
benfulcher
www.benfulcher.com
www.comp-engine.org
github.com/benfulcher/hctsa


Oliver Cliff


Annie Bryant


Trent Henderson

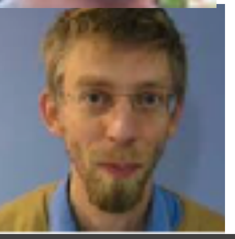
THE UNIVERSITY OF SYDNEY

https://dynamicsandneuralsystems.github.io/


Sarab Sethi


Carl Lubba

Imperial College London


Nick Jones

Selected References:

- Fulcher et al. (2013). Highly comparative time-series analysis: the empirical structure of time series and their methods. *J. Roy. Soc. Interface*.
- Fulcher & Jones (2014). Highly comparative feature-based time-series classification. *IEEE Trans. Knowl. Data Eng*.
- Fulcher & Jones (2017). *hctsa*: A Computational Framework for Automated Time-Series Phenotyping Using Massive Feature Extraction. *Cell Systems*.
- Cliff, Lizier, Tsuchiya, Fulcher (2022). Unifying Pairwise Interactions in Complex Dynamics. *arXiv* 220111941.
- Fulcher (2018). Feature-based time-series analysis, *Feature Engineering,* CRC Press.
- Lubba et al. (2020). catch22: CAnonical Time-series CHaracteristics. *Data Mining and Knowledge Discovery*.
- Fulcher et al. (2020). CompEngine: a self-organizing, living library of time-series data. *Scientific Data*.
- Henderson & Fulcher (2021). An Empirical Evaluation of Time-Series Feature Sets. In: *2021 International Conference on Data Mining Workshops (ICDMW)*.

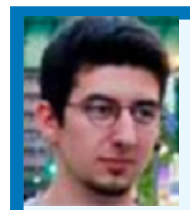
https://github.com/benfulcher/hctsa/wiki/Publications-using-hctsa

Quick demo of *hctsa*

Sample Dataset: https://github.com/benfulcher/hctsaTutorial_BonnEEG

Cloudstor for the input file and the pre-computed data

My generic advice for any dataset is to first run with *catch22* (and can scale up to *hctsa* later if needed)



catch22

Compute **22** time-series features

Fast-coded in C

Rcatch22 and **catch22.jl**

https://github.com/DynamicsAndNeuralSystems/catch22

hctsa

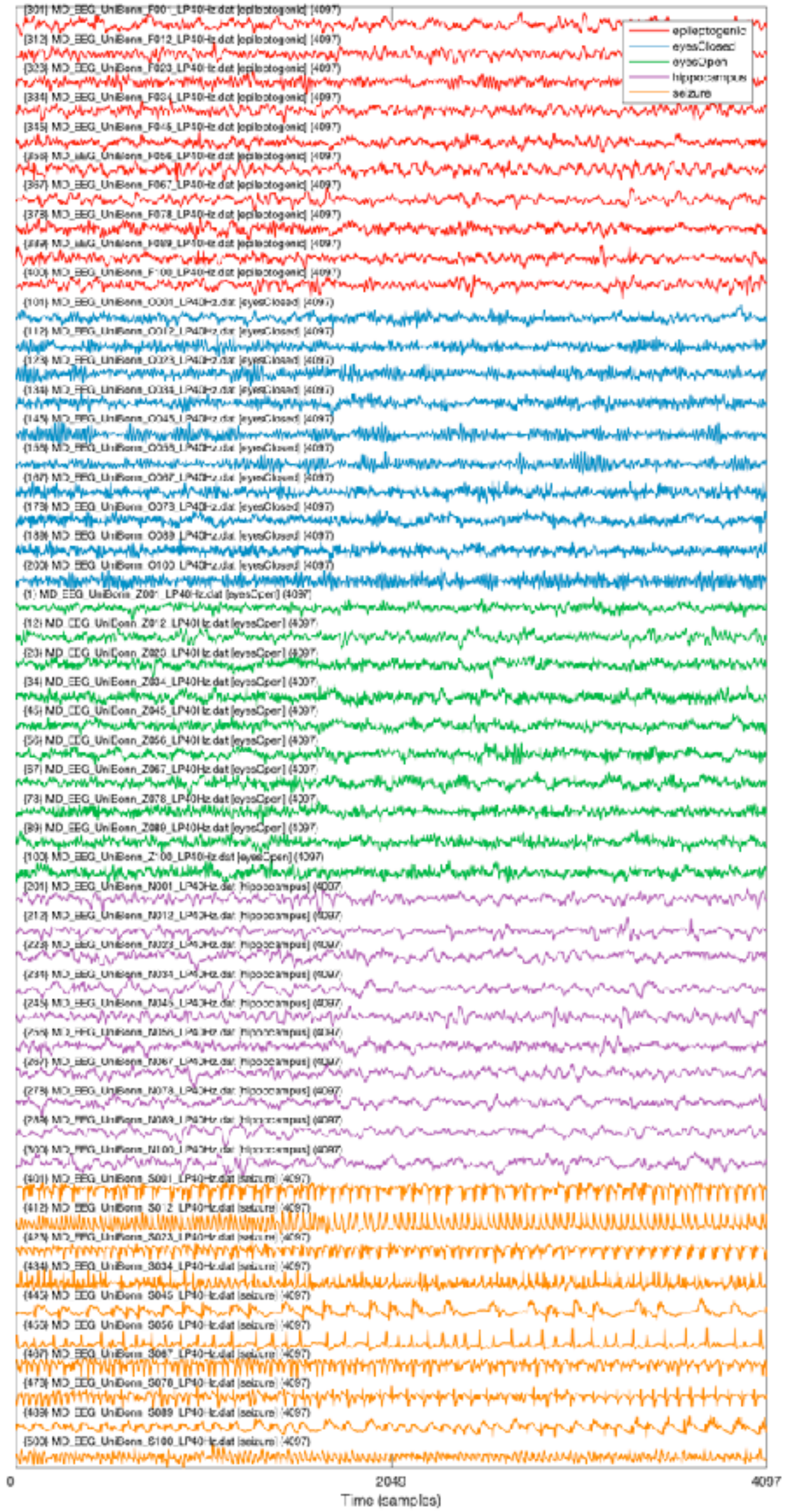
Compute >**7700** time-series features

Low-dimensional projections
Classification, …

https://github.com/benfulcher/hctsa

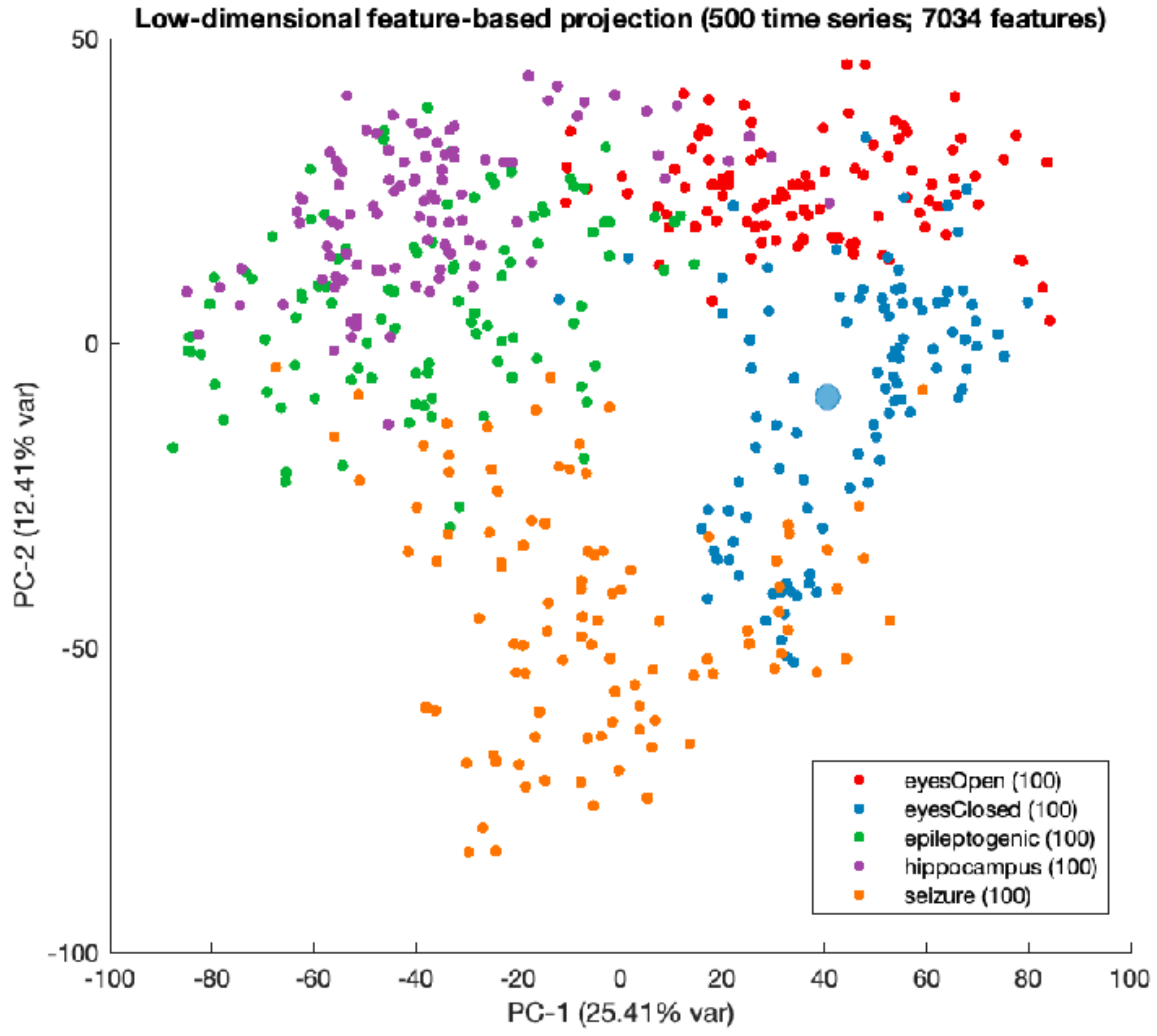
You want you to do this to your dataset!

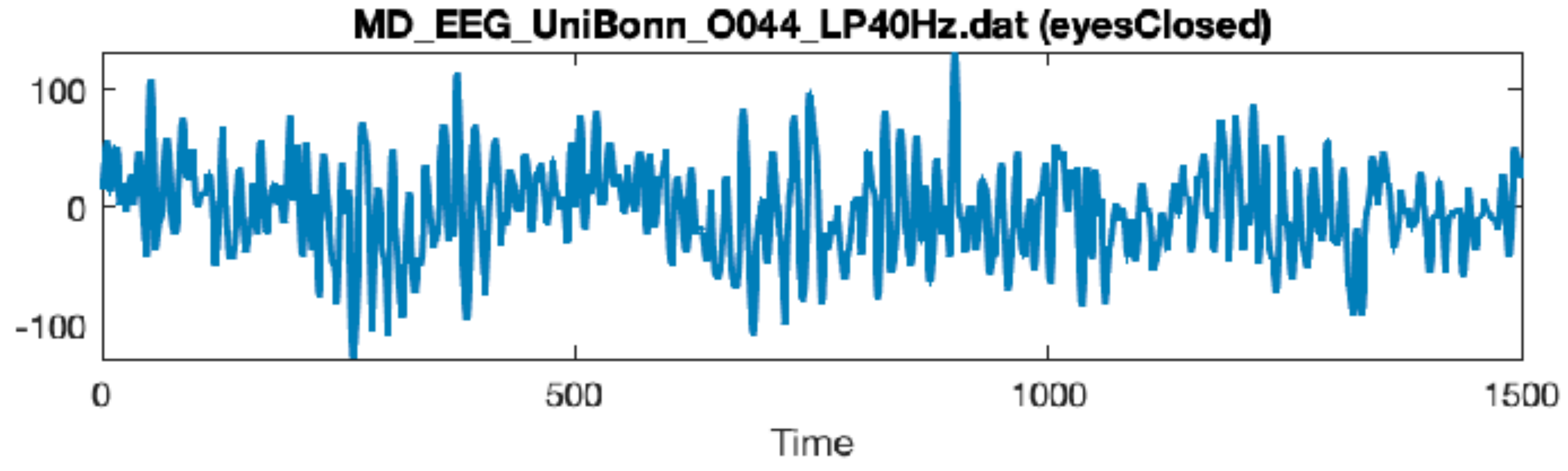
100 examples of each of 5 classes of EEG



hctsa

Interactive visualization



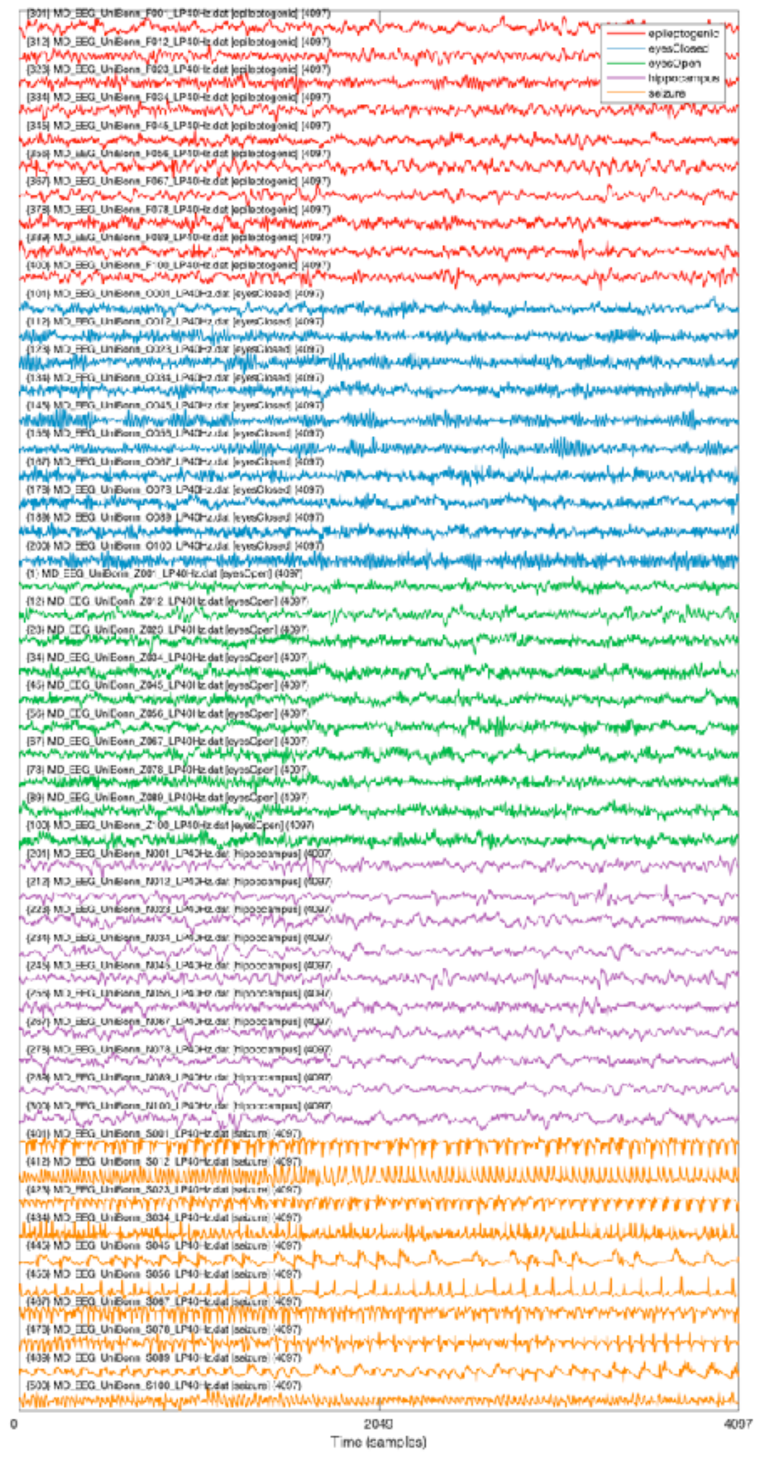


Demo

Load in a dataset → Compute time-series features → Interact with your low-dimensional data visualization

`TS_Init`

100 examples of each of 5 classes of EEG



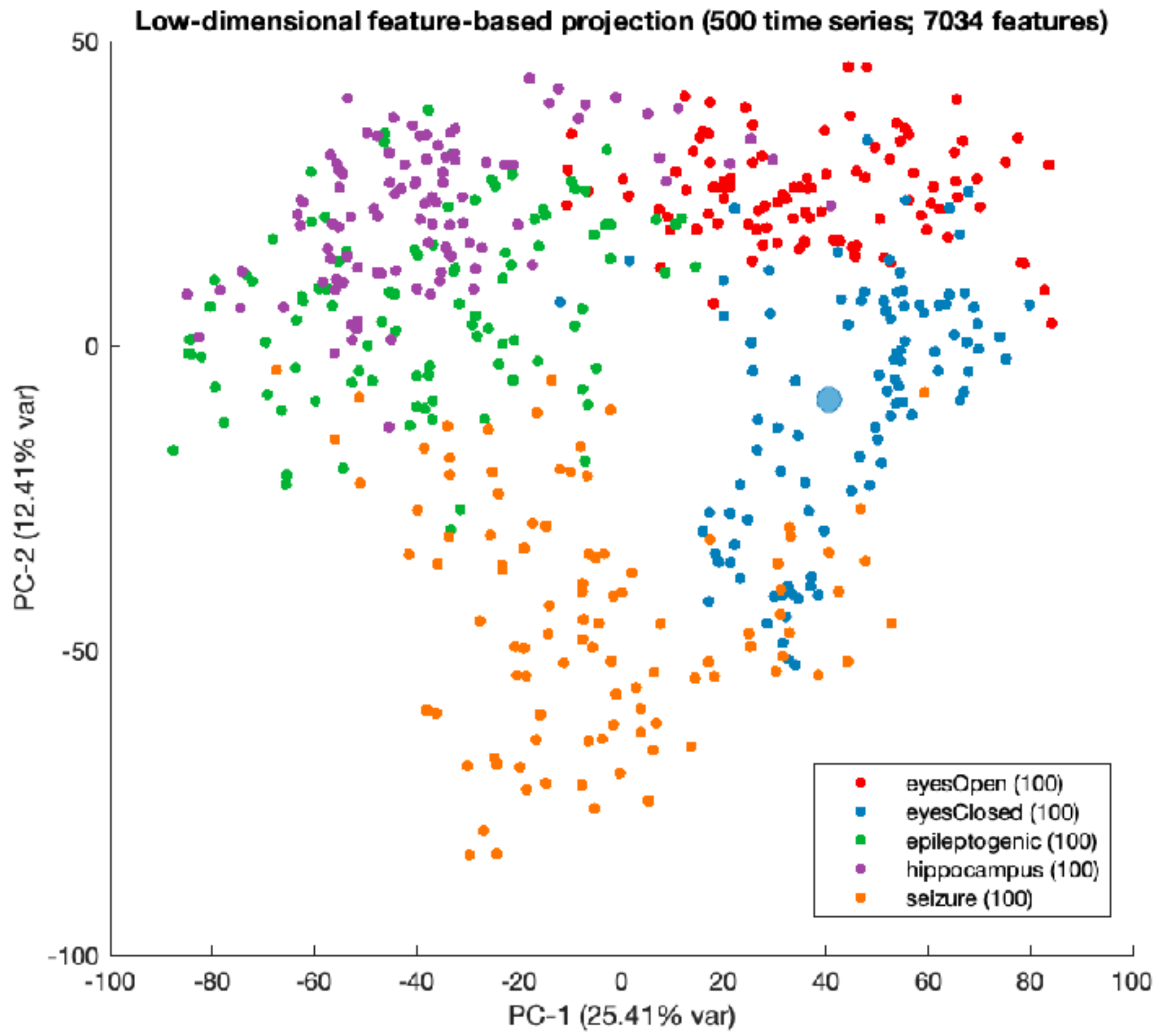
`TS_Compute`

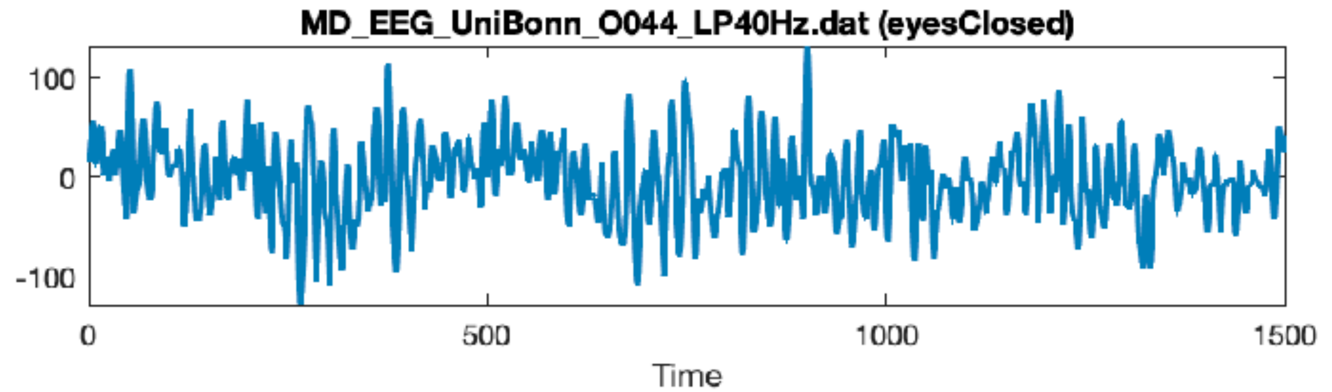
catch22 (22 features) for speed
hctsa (>7k) for comprehensiveness

`TS_Normalize`

Put all features on a similar scale

`TS_LowDimInspect`





https://github.com/benfulcher/hctsaTutorial_BonnEEG

1 Prepare Dataset:

QuickInteractiveVis.mlx

`INP_Bonn_EEG.mat`

- `labels` 500 x 1 cell strings uniquely identify each time series
- `timeSeriesData` 500 x 1 cell vectors of time-series data
- `keywords` 500 x 1 cell class labels

2

Initialize (default *hctsa* feature set): `TS_Init('INP_Bonn_EEG.mat')`

Initialize (catch22 feature set): `TS_Init('INP_Bonn_EEG.mat','INP_mops_catch22.txt','INP_ops_catch22.txt',true)`

Generates: `HCTSA.mat`

- `TS_DataMat` 500 (time series) x 22 (features) matrix [empty]
- `TimeSeries` 500-row table with information about time series
- `Operations` 22-row table with information about operations/features

---

3

Compute all features (without parallelization): `TS_Compute(false);`

(very fast for *catch22*)

4

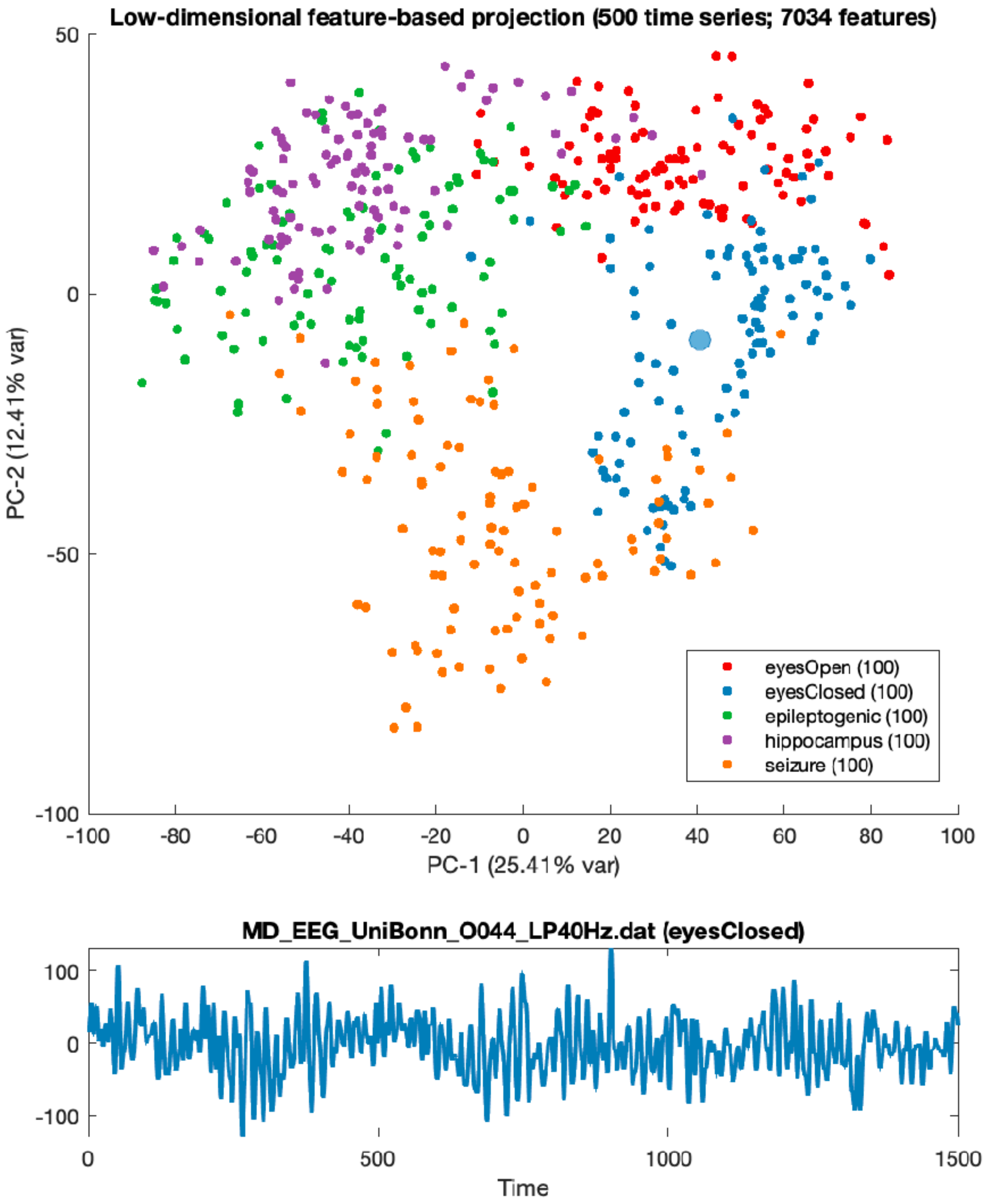
Label Groups and normalize features to a similar scale (and filter poor performers):

```
TS_LabelGroups();
TS_Normalize();
```

5

Visualize! Analyze! E.g., Play with a low-dimensional representation!: `TS_LowDimInspect();`

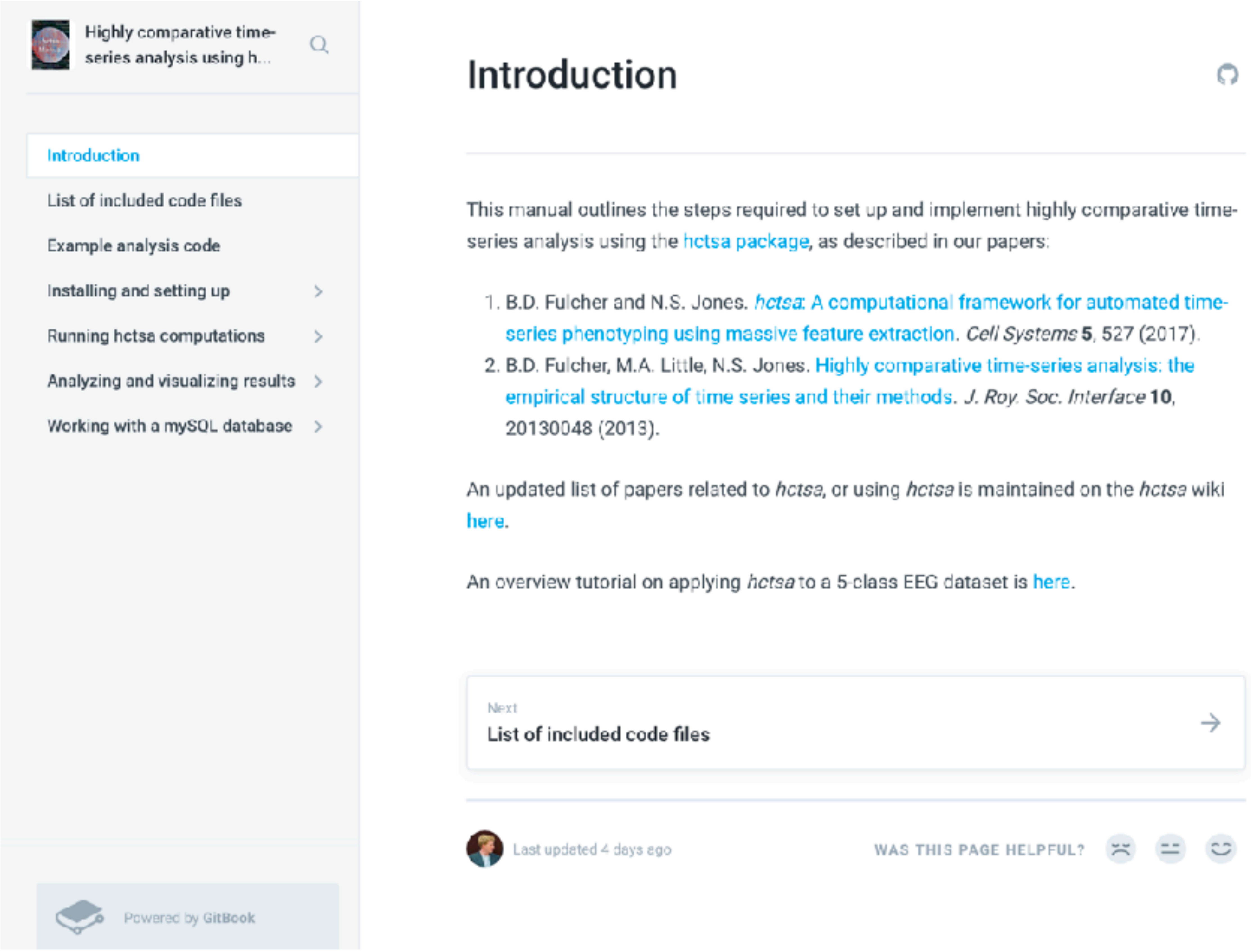
(Many other visualizations: see https://github.com/benfulcher/hctsaTutorial_BonnEEG)



Low-dimensional feature-based projection (500 time series; 7034 features)
PC-2 (12.41% var)
PC-1 (25.41% var)
50
0
-50
-100
-100
-80
-60
-40
-20
0
20
40
60
80
100
eyesOpen (100)
eyesClosed (100)
epileptogenic (100)
hippocampus (100)
seizure (100)
MD_EEG_UniBonn_O044_LP40Hz.dat (eyesClosed)
100
0
-100
0
500
1000
1500
Time

Going Further

Comprehensive documentation on GitBook + wiki



https://hctsa-users.gitbook.io/hctsa-manual/

https://github.com/benfulcher/hctsa/wiki

- How accurately can I classify?
- What types of time-series properties distinguish the classes?

Work through the full suite of *hctsa* functionality for this dataset:

https://github.com/benfulcher/hctsaTutorial_BonnEEG

Work through other *hctsa* analyses for fly and worm phenotyping (open code and pre-computed data):

https://github.com/benfulcher/hctsa_phenotypingFly

https://github.com/benfulcher/hctsa_phenotypingWorm