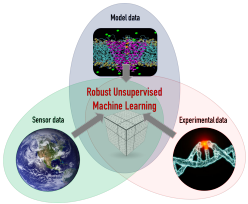
Novel Unsupervised Machine Learning Methods for Extraction of Features Characterizing Datasets and Models

Velimir V. Vesselinov (monty) (vvv@lanl.gov)
Computation Earth Sciences, Los Alamos National Laboratory, NM, USA



AGU Fall Meeting 2018

LA-UR-18-31366

- We have developed a series of novel unsupervised Machine Learning (ML) methods
- Our unique ML methods are based in matrix/tensor factorization coupled with custom k-means clustering and nonnegativity/sparsity constraints:
  - **NMF**k: Nonnegative **Matrix** Factorization
  - **NTF**k: Nonnegative **Tensor** Factorization
- **NMF**k / **NTF**k are capable to efficiently process large datasets (GB/TB's) utilizing GPU's & TPU's (TensorFlow, PyTorch, MXNet)
- **NMF**k / **NTF**k have been applied to analyze a series of real-world analyses

Why Unsupervised Machine Learning (ML)?

- **Supervised** ML: requires prior categorization (knowledge) of the processed data

  Example: Recognize images of cats and dogs after extensive training; but cannot recognize horses if not trained

  Cannot find something that we do not already know

- **Unsupervised** ML: extracts hidden features (signals) in the processed data without any prior information (**exploratory analysis** for **data-driven science**)

  Example: Identify features that distinguish images of animals (e.g., cats, dogs, horses, etc.); without prior information or training

Why Unsupervised Machine Learning (ML)?

- **Supervised** ML: requires prior categorization (knowledge) of the processed data

  Example: Recognize images of cats and dogs after extensive training; but cannot recognize horses if not trained

  Cannot find something that we do not already know

- **Unsupervised** ML: extracts hidden features (signals) in the processed data without any prior information (**exploratory analysis** for **data-driven science**)

  Example: Identify features that distinguish images of animals (e.g., cats, dogs, horses, etc.); without prior information or training

Why Unsupervised Machine Learning (ML)?

- **Supervised** ML: requires prior categorization (knowledge) of the processed data

  Example: Recognize images of cats and dogs after extensive training; but cannot recognize horses if not trained
  Cannot find something that we do not already know

- **Unsupervised** ML: extracts hidden features (signals) in the processed data without any prior information (**exploratory analysis** for **data-driven science**)

  Example: Identify features that distinguish images of animals (e.g., cats, dogs, horses, etc.); without prior information or training

Why Unsupervised Machine Learning (ML)?

- **Supervised** ML: requires prior categorization (knowledge) of the processed data

  Example: Recognize images of cats and dogs after extensive training; but cannot recognize horses if not trained
  Cannot find something that we do not already know

- **Unsupervised** ML: extracts hidden features (signals) in the processed data without any prior information (**exploratory analysis** for **data-driven science**)

  Example: Identify features that distinguish images of animals (e.g., cats, dogs, horses, etc.); without prior information or training

Why Unsupervised Machine Learning (ML)?

- **Supervised** ML: requires prior categorization (knowledge) of the processed data

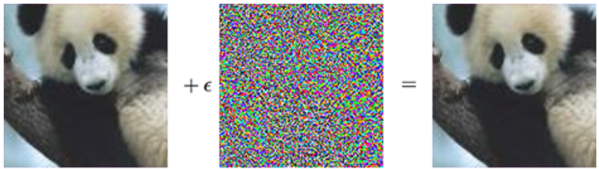
  Example: Recognize images of cats and dogs after extensive training; but cannot recognize horses if not trained
  Cannot find something that we do not already know

- **Unsupervised** ML: extracts hidden features (signals) in the processed data without any prior information (**exploratory analysis** for **data-driven science**)

  Example: Identify features that distinguish images of animals (e.g., cats, dogs, horses, etc.); without prior information or training

Why not supervised Machine Learning (ML)

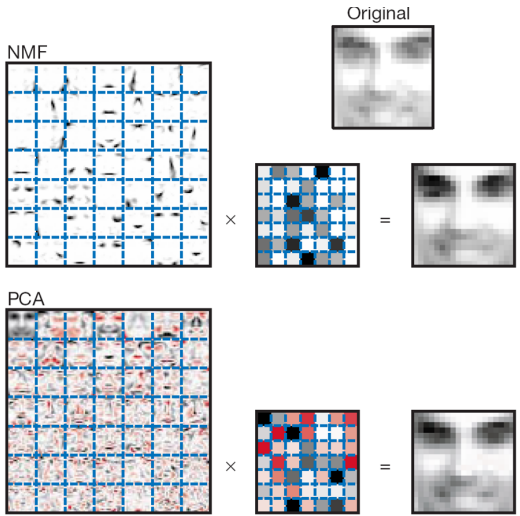
- **Supervised** ML
  - can introduce subjectivity (through the labeling process)
  - does not provide insights why horses are different than dogs / cats
  - cannot make predictions
  - requires huge training (labeled) datasets
  - is impacted by “adversarial examples”



$\Rightarrow$ major limitations of the **supervised** methods
for **data-analytics** and **data-driven science** applications

Why nonnegativity?

- NMF vs PCA (Lee & Seung, 1999)
- NMF: Nonnegative Matrix Factorization
- PCA: Principal Component Analysis

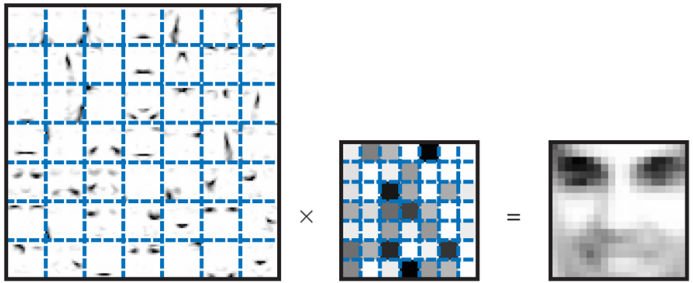


Nonnegativity constraints provide meaningful and interpretable results (+sparsity)

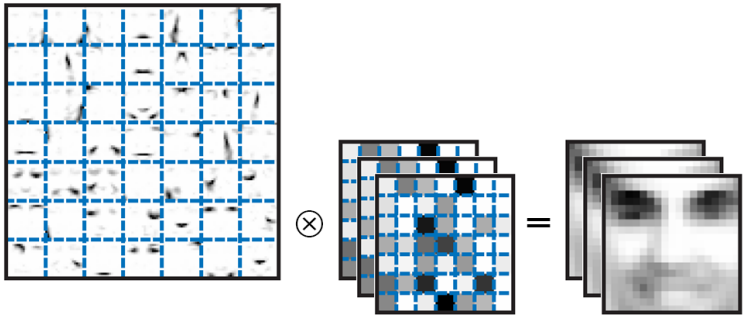
Why tensors?

- **Tensors** (multi-dimensional arrays / multi-modal data) are everywhere:
  - color image is a 3-D tensor (RGB)
  - color movie is a 4-D tensor (RGB + time)
  - observable data are typically a 5-D tensor (x, y, z, t, scalars...)
  - model outputs are typically a 5-D tensor (x, y, z, t, scalars...)
  - n model parameters (e.g., conductivity, capacity, etc.) impacting model outputs form a $(n+5)$-D tensor
  - n parameters (e.g., pressure, temperature, pH, species concentrations, etc.) impacting experiments (e.g. reaction rate) form a n-D tensor

NMF: Nonnegative Matrix Factorization



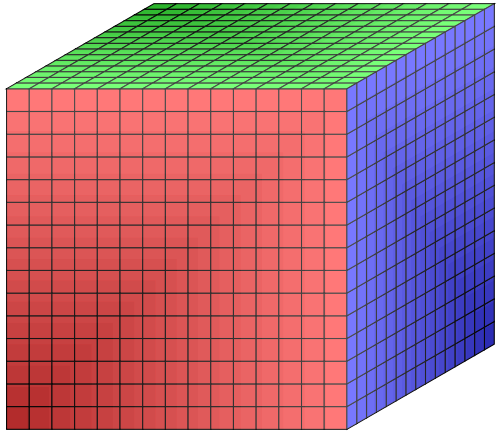
NTF: Nonnegative Tensor Factorization



Why not use PCA/SVD?

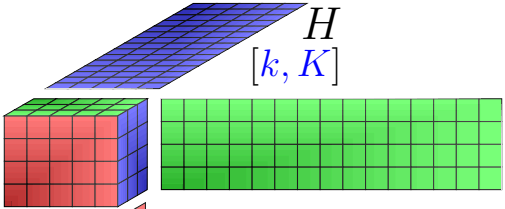
- There is no direct equivalent of PCA/SVD for multi-dimensional arrays (tensors)

Tucker Tensor Factorization (3D case): Tucker-3



X
$[K, M, N]$

$\approx$



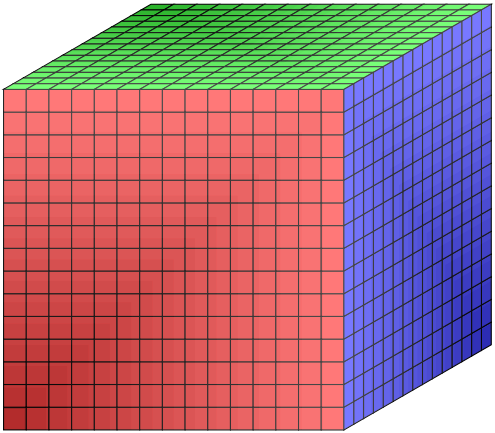
H
$[k, K]$

G
$[k, m, n]$

W
$[n, N]$

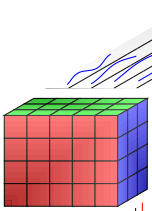
V
$[m, M]$

$$X \approx G \otimes H \otimes W \otimes V$$

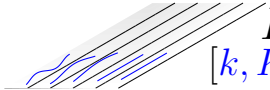


$$X \\ [K, M, N]$$

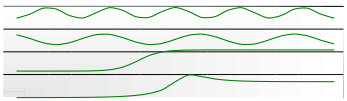
$\approx$



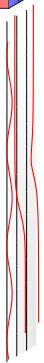
$$G \\ [k, m, n]$$



$$H \\ [k, K]$$



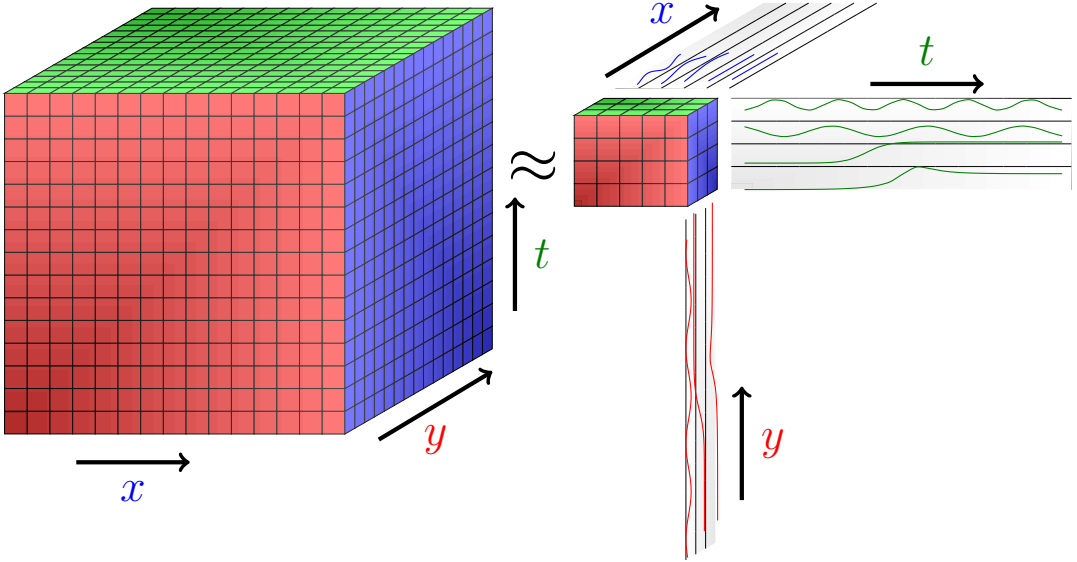
$$W \\ [n, N]$$



$$V \\ [m, M]$$

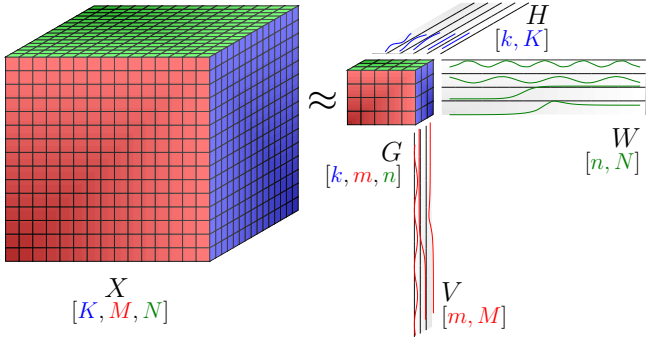
$$X \approx G \otimes H \otimes W \otimes V$$

Tucker Tensor Decomposition: Feature extraction



Tucker Tensor Decomposition: Feature extraction

- Tucker decomposition is achieved through minimization
- Nonnegativity and sparsity constraints help the feature extraction
- Optimal number of features $[k, m, n]$ is estimated through k-means clustering of a series minimization solutions with random initial guesses

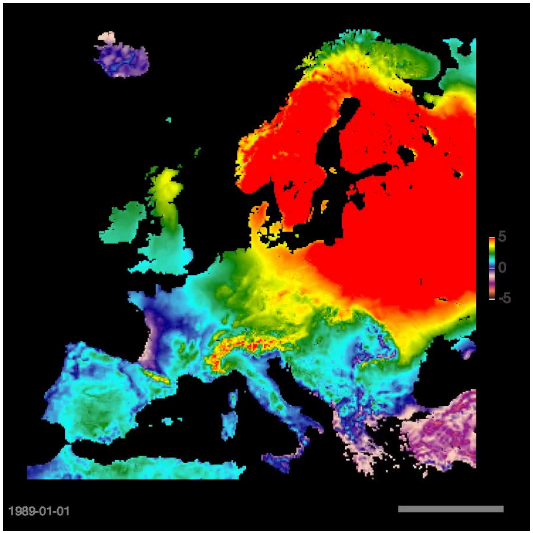


- **Field Data**:
  - Groundwater contaminant migration
  - US Climate
  - Geothermal
  - Seismic
- **Lab Data**:
  - X-ray Spectroscopy
  - UV Fluorescence Spectroscopy
  - Microbial population analyses
- **Operational Data**:
  - LANSCE: Los Alamos Neutron Accelerator
  - Hydrocarbon (oil/gas) production
- **Model Data**:
  - Reactive mixing $A + B \rightarrow C$
  - Phase separation of co-polymers
  - Molecular Dynamics of proteins
  - Lattice-Boltzmann simulations of fluid displacement
  - Europe Climate modeling

- Vesselinov, Munuduru, Karra, O'Maley, Alexandrov, Unsupervised Machine Learning Based on Non-Negative Tensor Factorization for Analyzing Reactive-Mixing, **Journal of Computational Physics**, (in review), 2018.
- Vesselinov, O'Malley, Alexandrov, Nonnegative Tensor Factorization for Contaminant Source Identification, **Journal of Contaminant Hydrology**, (accepted), 2018.
- Stanev, Vesselinov, Kusne, Antoszewski, Takeuchi, Alexandrov, Unsupervised Phase Mapping of X-ray Diffraction Data by Nonnegative Matrix Factorization Integrated with Custom Clustering, **Nature Computational Materials**, 2018.
- O'Malley, Vesselinov, Alexandrov, Alexandrov, Nonnegative/binary matrix factorization with a D-Wave quantum annealer, **PLOS ONE**, (accepted), 2018.
- Vesselinov, O'Malley, Alexandrov, Contaminant source identification using semi-supervised machine learning, **Journal of Contaminant Hydrology**, 10.1016/j.jconhyd.2017.11.002, 2017.
- Alexandrov, Vesselinov, Blind source separation for groundwater level analysis based on nonnegative matrix factorization, **WRR**, 10.1002/2013WR015037, 2014.

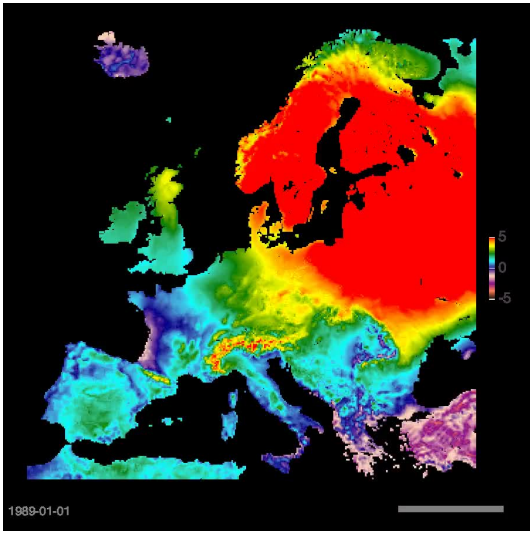
Climate model of Europe

- **CLM/ParFlow** model developed by Stefan Kollet, Carina Furusho, Klaus Görgen et al. (Forschungszentrum Jülich, Germany)

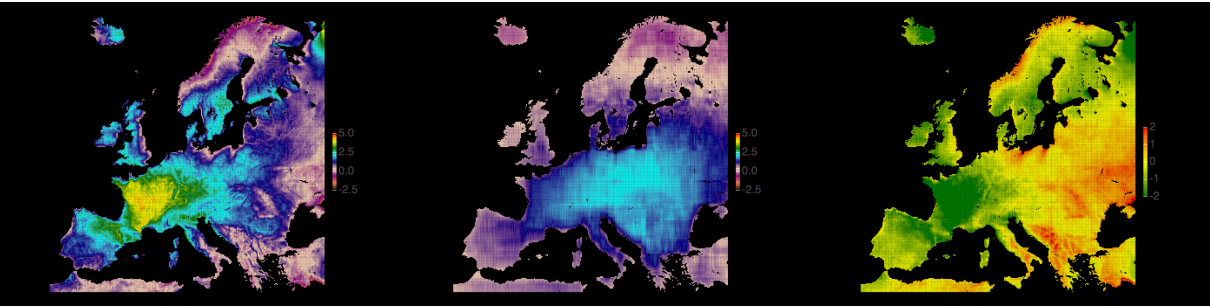


Climate model of Europe: air temperature 1989-2017

- monthly fluctuations in the air temperature from 1989 to 2017 [$^{\circ}C$]
- Tensor: $(316 \times 316 \times 348)$ $(columns \times rows \times months)$
- **NTF**k applied to extract dominant hidden (latent) features based on spatial footprints and temporal characteristics

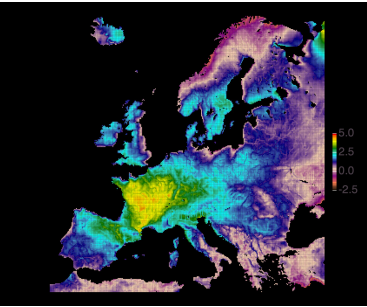


Climate model of Europe: 2003 air temperature reconstruction by 3 features

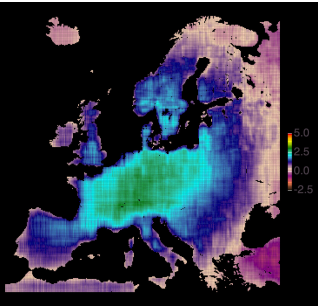


Climate model of Europe: 2003 air temperature reconstruction by 4 features

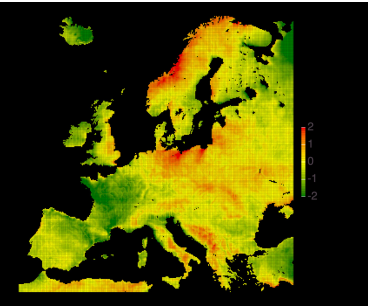
Original



Reconstruction

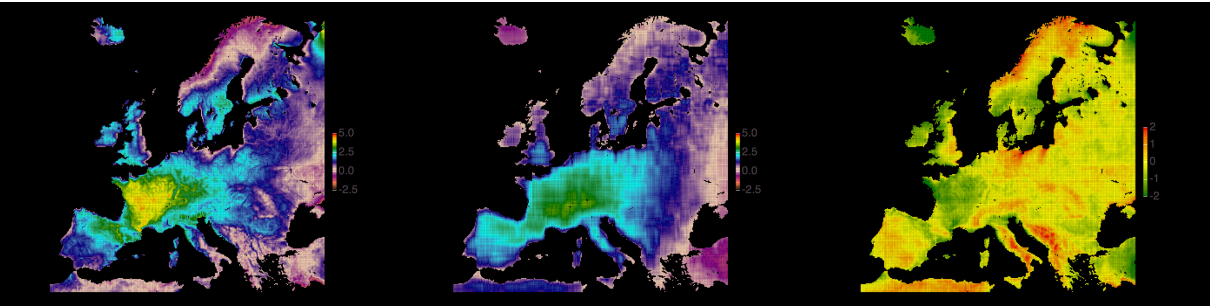


Error



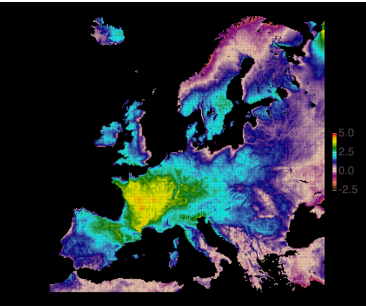
Climate model of Europe: 2003 air temperature reconstruction by 5 features

Original | Reconstruction | Error

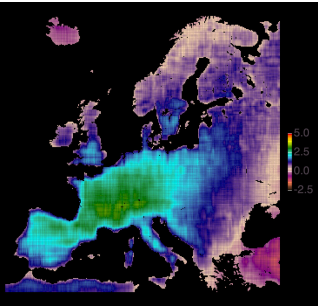


Climate model of Europe: 2003 air temperature reconstruction by 6 features

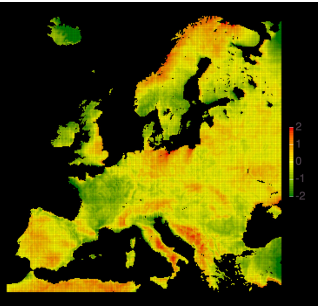
Original



Reconstruction

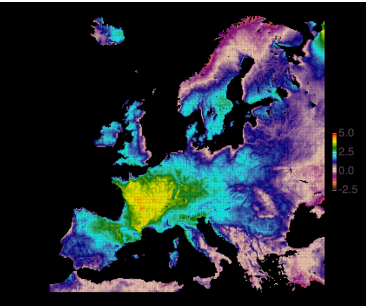


Error

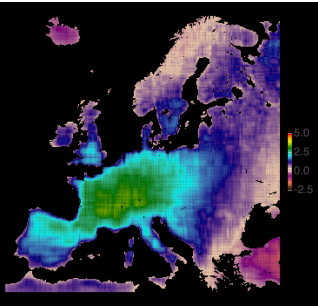


Climate model of Europe: 2003 air temperature reconstruction by 7 features

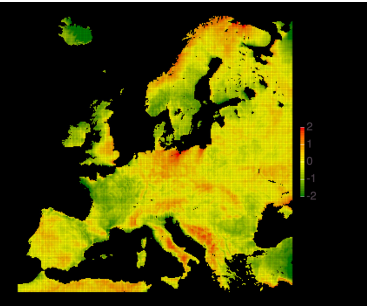
Original



Reconstruction

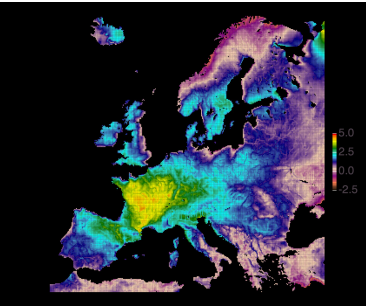


Error

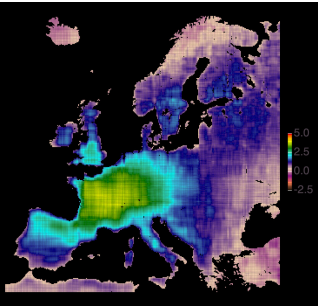


Climate model of Europe: 2003 air temperature reconstruction by 8 features

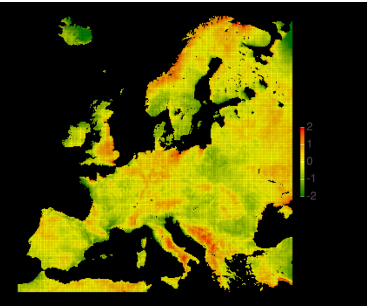
Original



Reconstruction

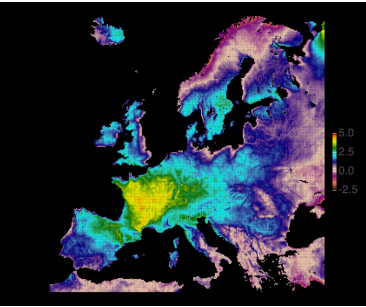


Error

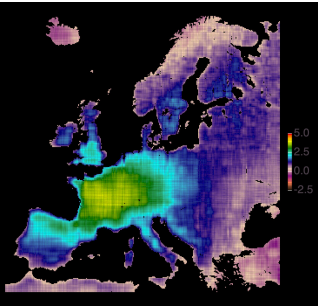


Climate model of Europe: 2003 air temperature reconstruction by 9 features

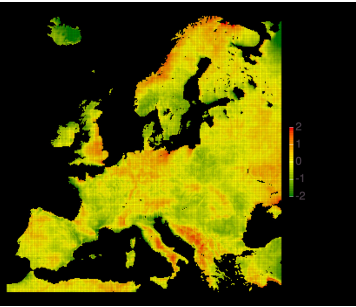
Original



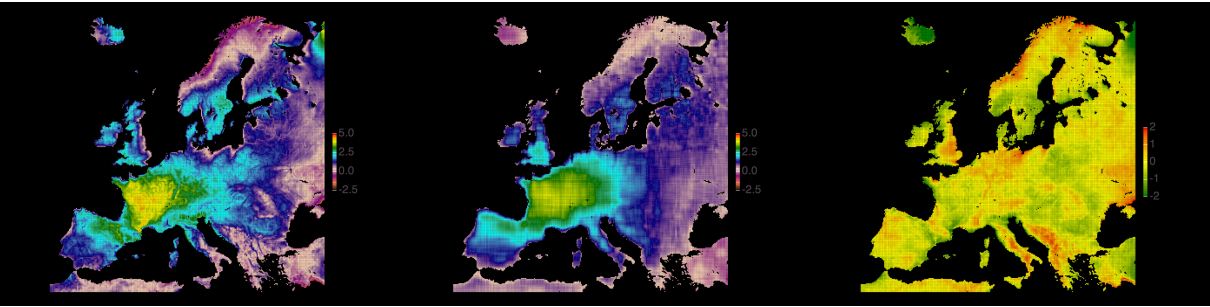
Reconstruction



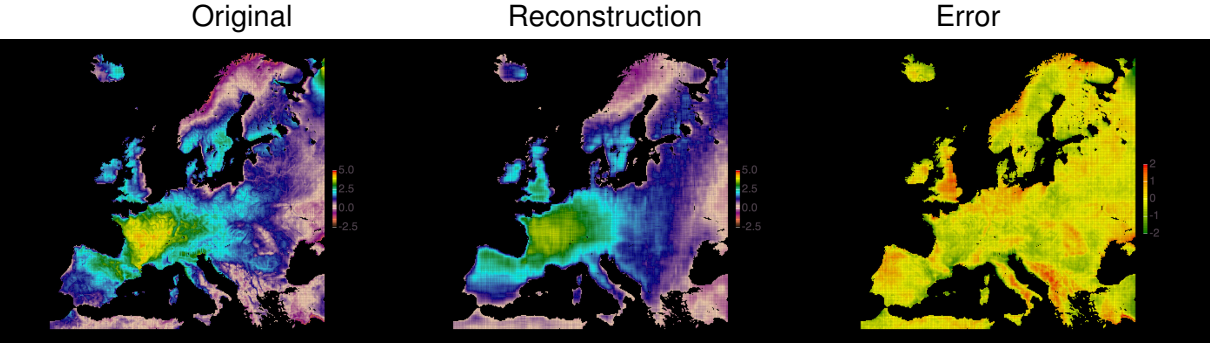
Error



Climate model of Europe: 2003 air temperature reconstruction by 10 features

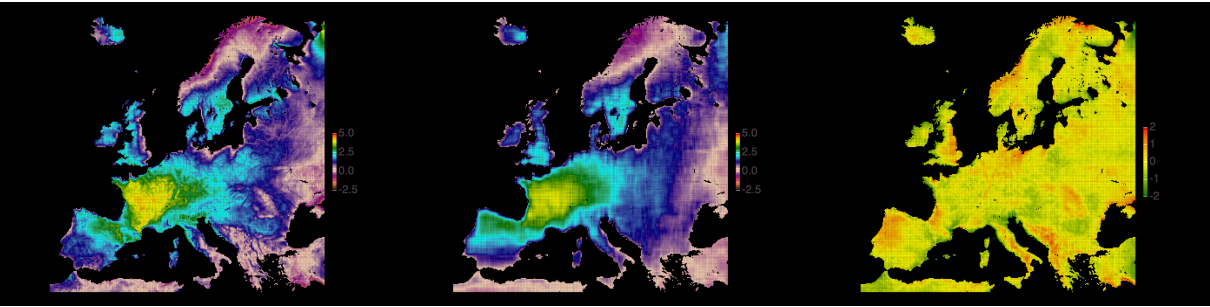


Climate model of Europe: 2003 air temperature reconstruction by 15 features

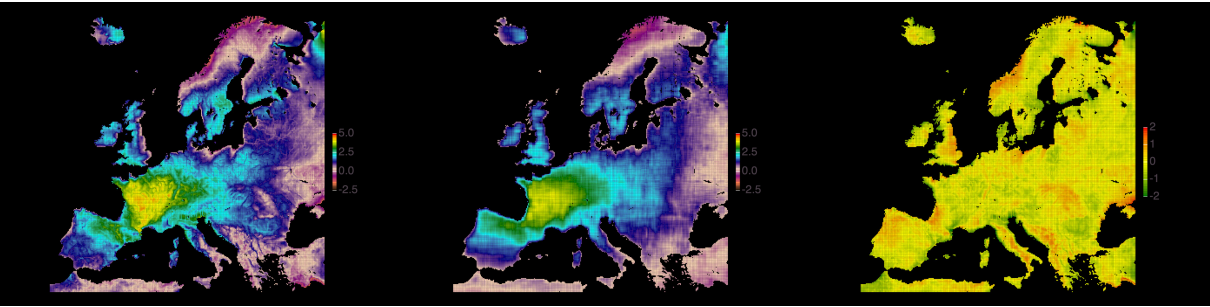


Climate model of Europe: 2003 air temperature reconstruction by 20 features

Original | Reconstruction | Error

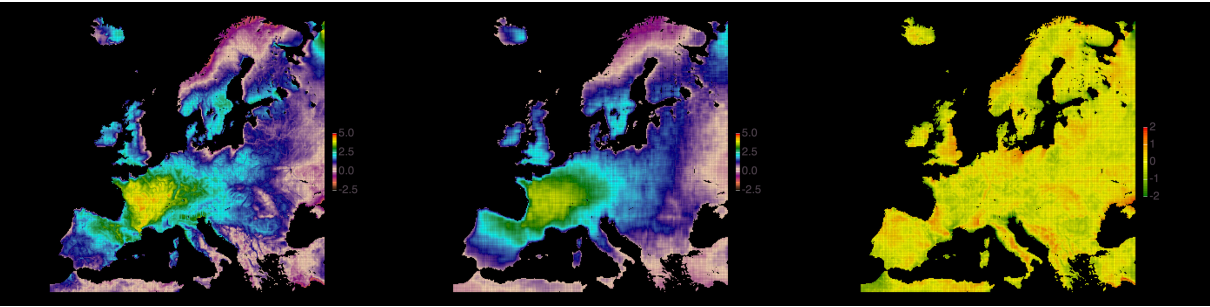


Climate model of Europe: 2003 air temperature reconstruction by 25 features



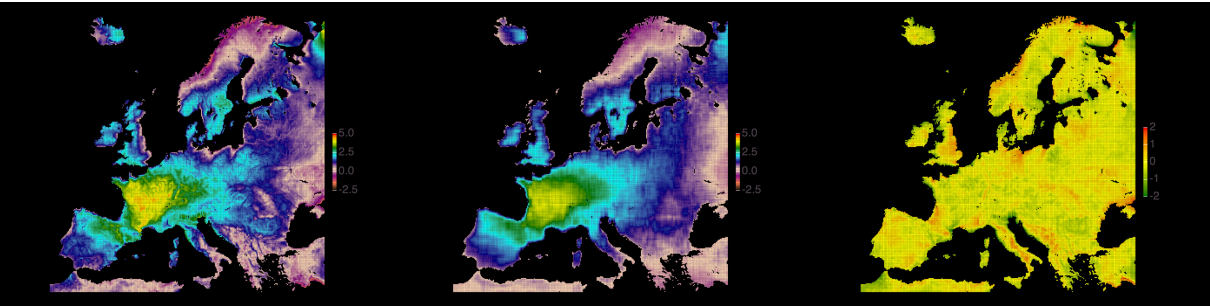
Climate model of Europe: 2003 air temperature reconstruction by 30 features

Original | Reconstruction | Error



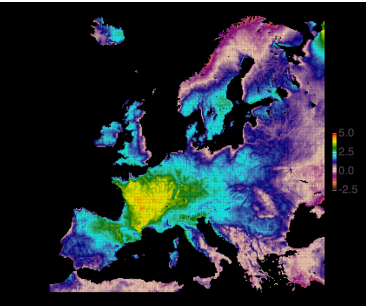
Climate model of Europe: 2003 air temperature reconstruction by 35 features

Original | Reconstruction | Error

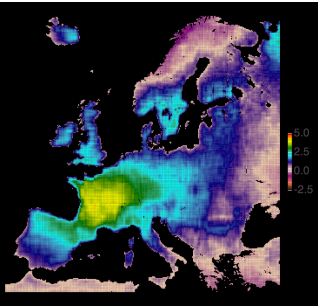


Climate model of Europe: 2003 air temperature reconstruction by 40 features

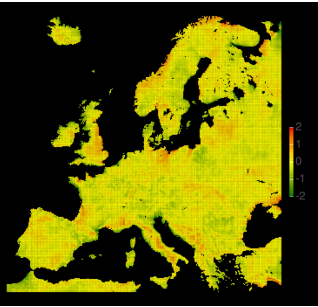
Original



Reconstruction

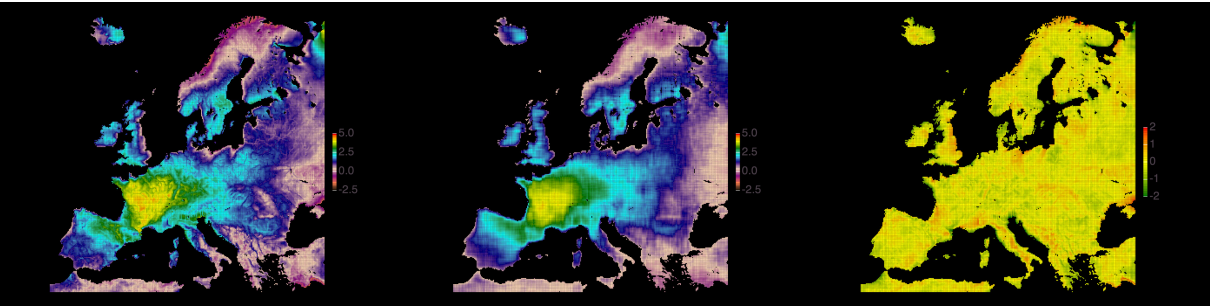


Error

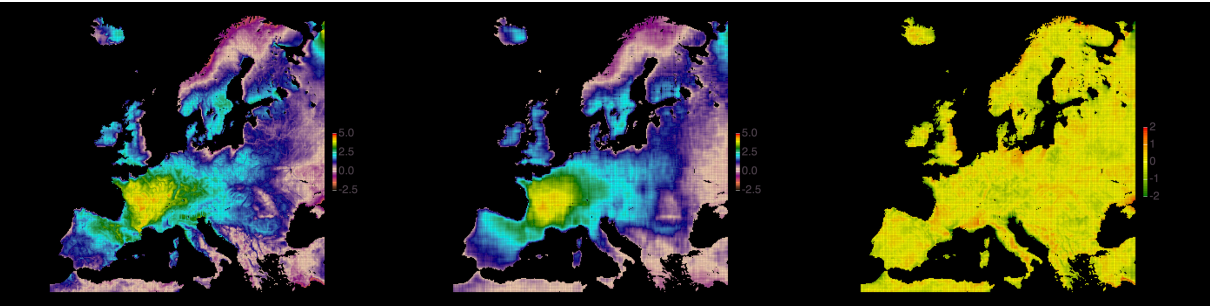


Climate model of Europe: 2003 air temperature reconstruction by 45 features

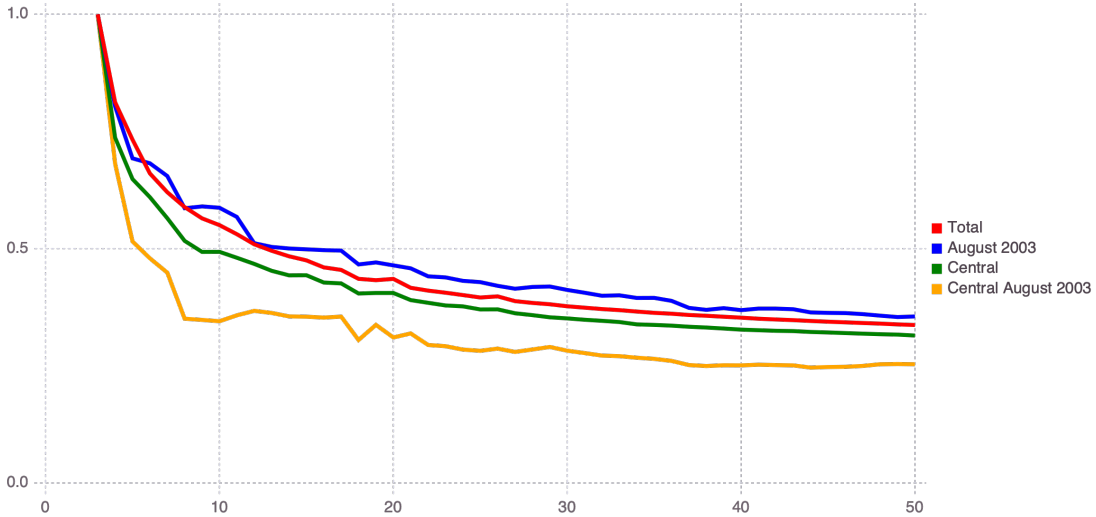
Original | Reconstruction | Error



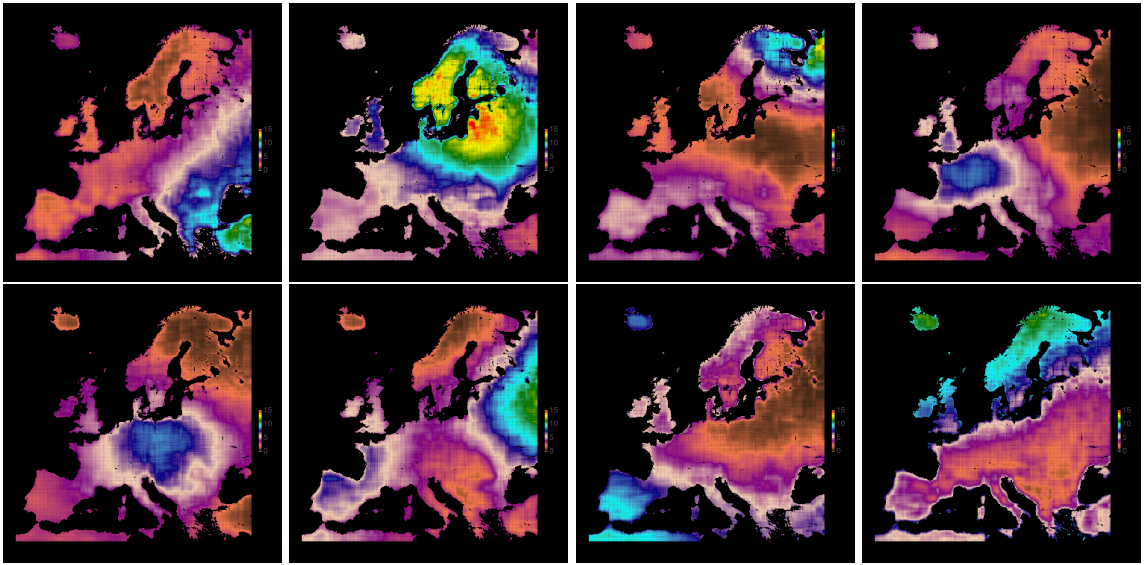
Climate model of Europe: 2003 air temperature reconstruction by 50 features



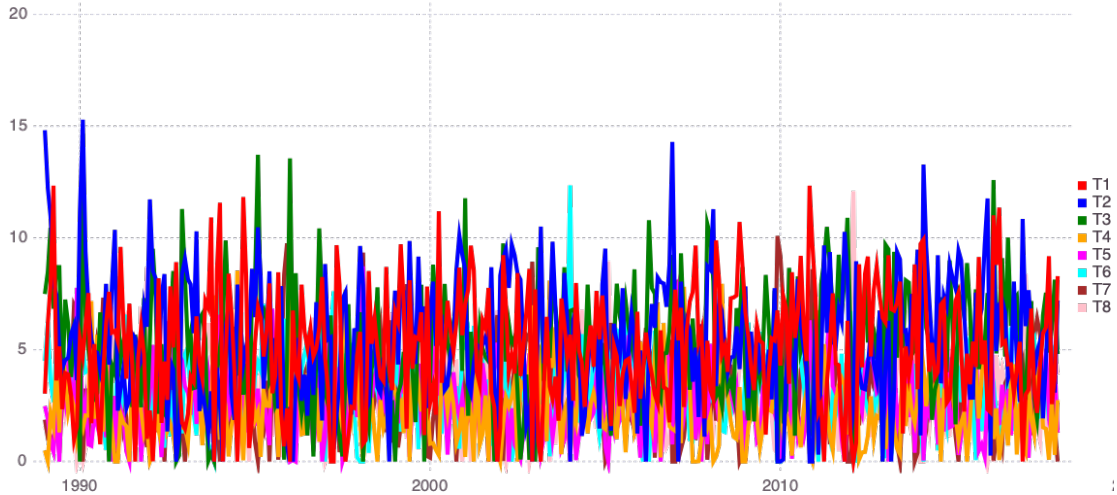
Climate model of Europe: air temperature reconstruction errors



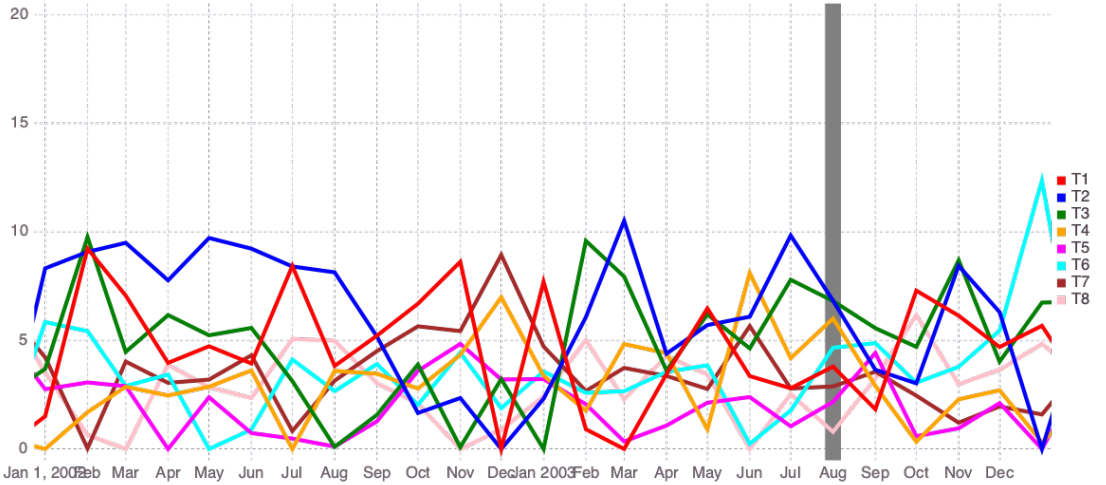
Climate model of Europe: air temperature features (8)



Climate model of Europe: air temperature features (8) 1989-2017



Climate model of Europe: air temperature features (8) 2002-2003

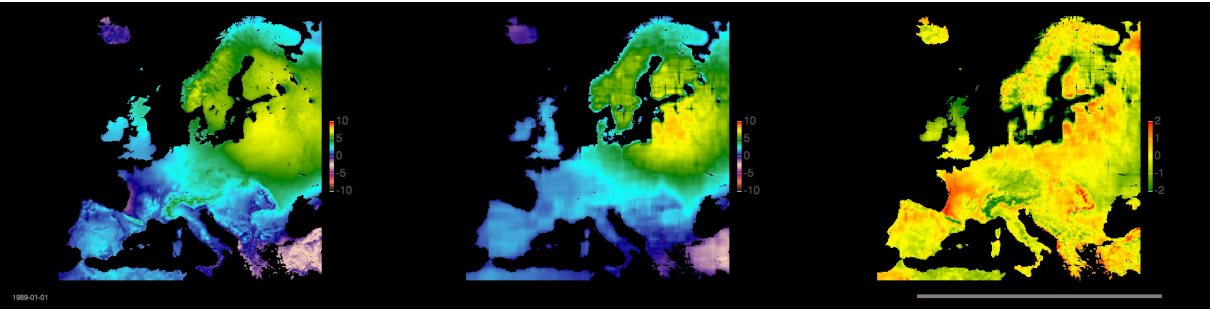


Climate model of Europe: air temperature reconstruction by 8 features

Original

Reconstruction

Error

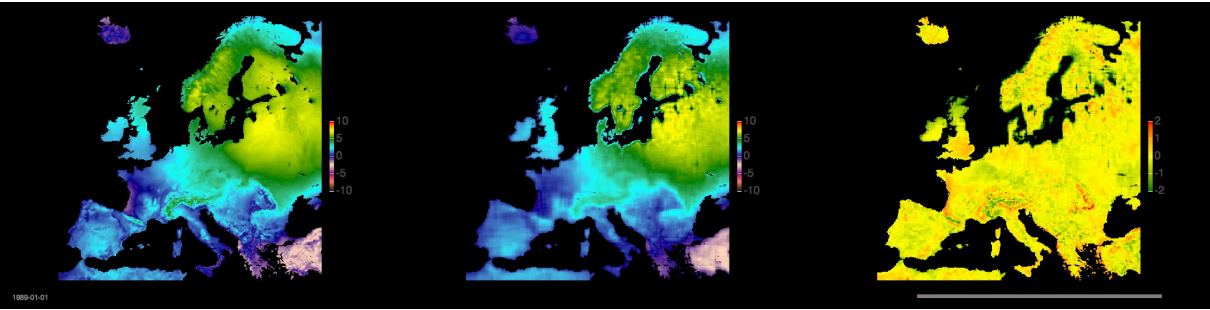


Climate model of Europe: air temperature reconstruction by 50 features

Original

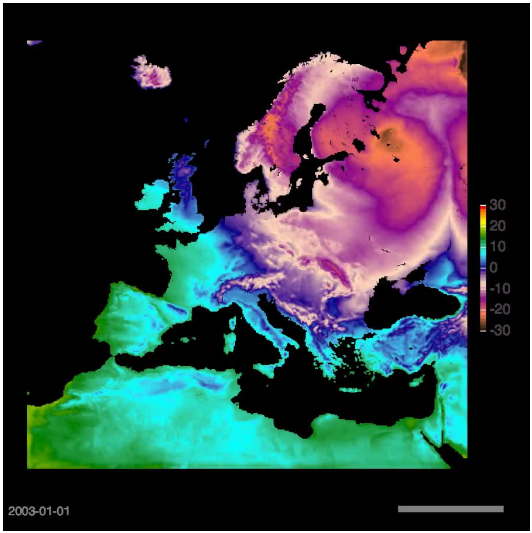
Reconstruction

Error

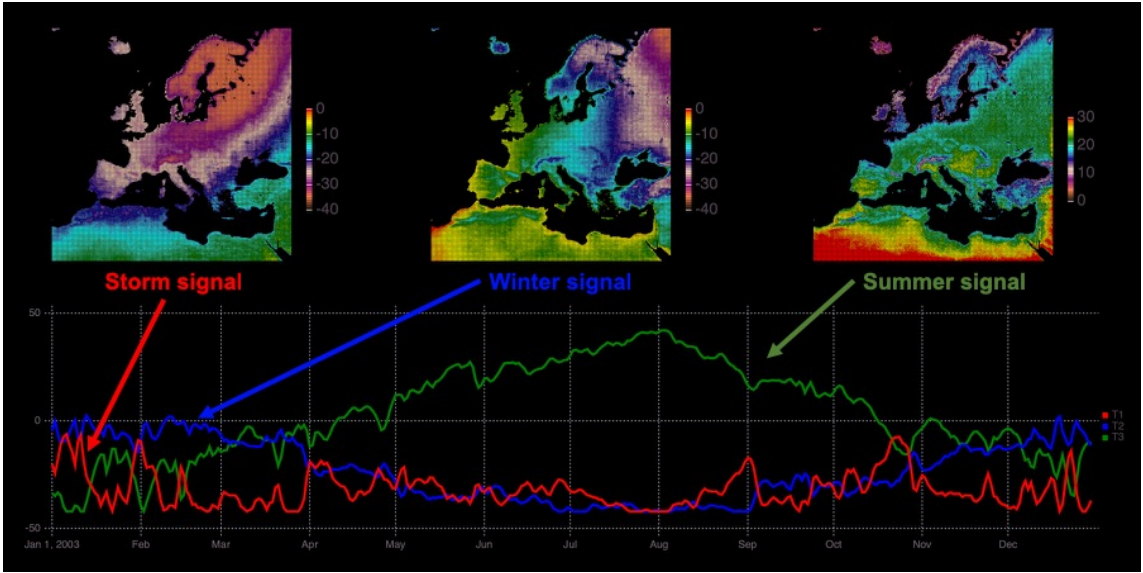


Climate model of Europe: 2003 air temperature

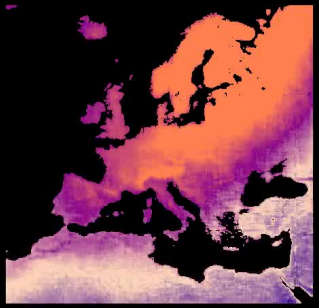
- daily fluctuations in the air temperature $[^{\circ}C]$
- Tensor: $(424 \times 412 \times 365)$
  $(columns \times rows \times days)$
- **NTF**k applied to extracts hidden features

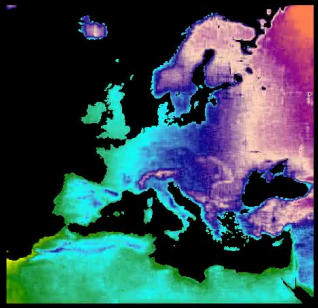


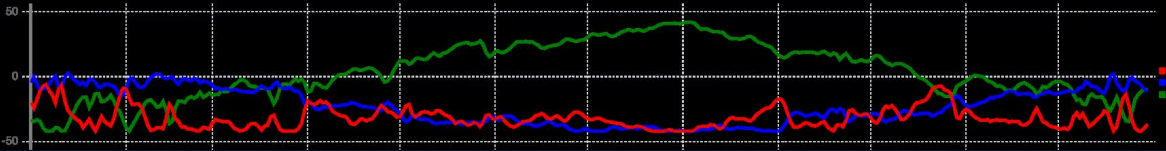
Climate model of Europe: 2003 temperature fluctuations represented by 3 features



Climate model of Europe: 2003 temperature fluctuations represented by 3 features

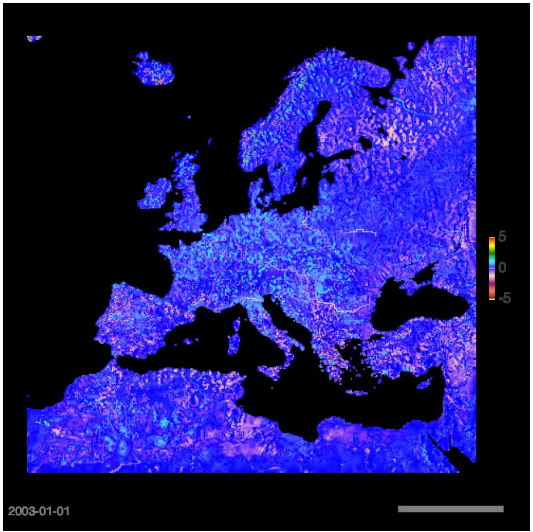




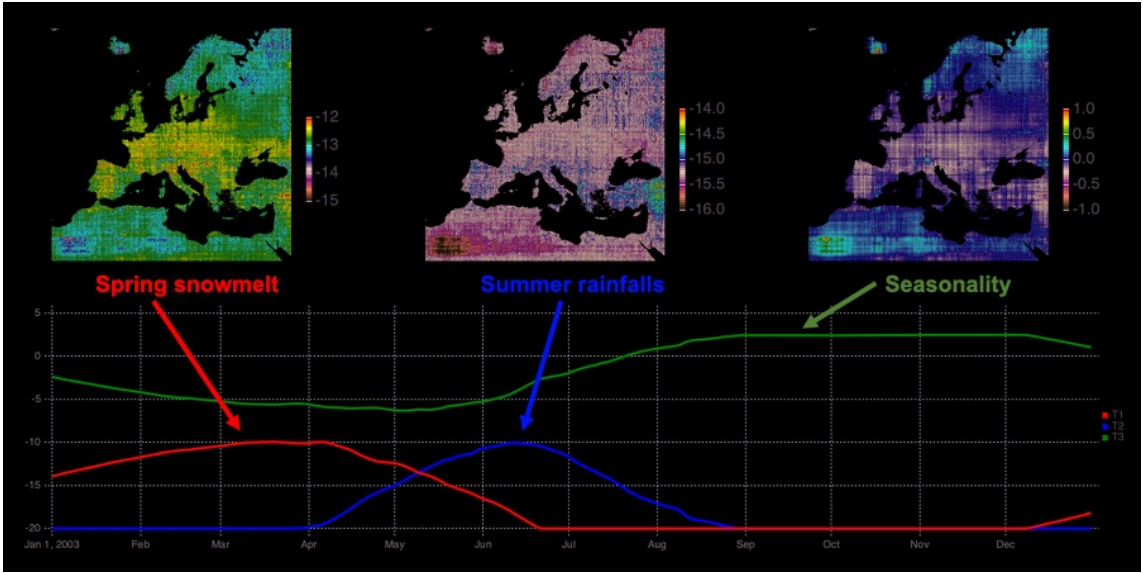


Climate model of Europe: 2003 water-table depth

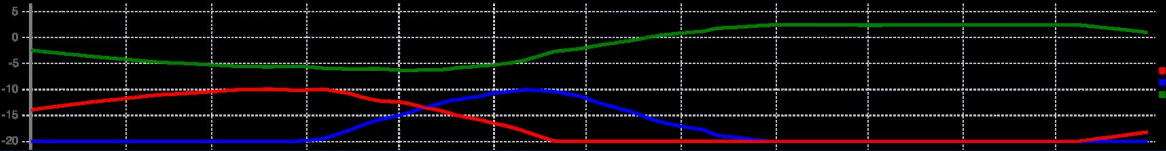
- fluctuations in the water-table depth $[m]$
- Tensor: $(424 \times 412 \times 365)$
  $(columns \times rows \times days)$
- **NTF**k extracts spatial and temporal footprints of dominant features



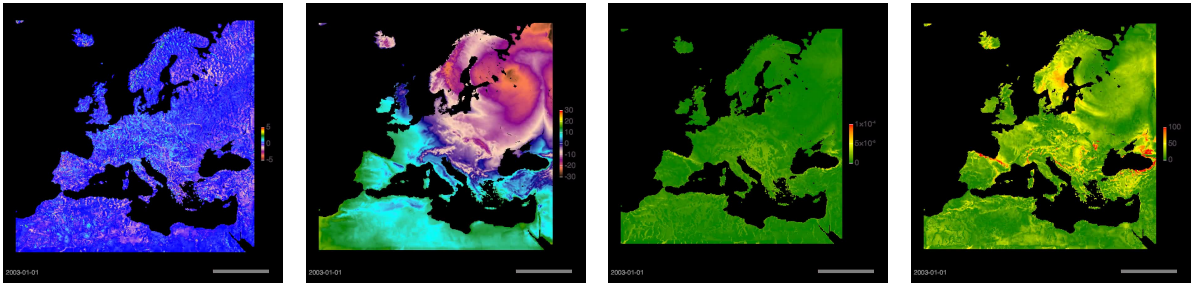
Climate model of Europe: 2003 water-table fluctuations represented by 3 features



Climate model of Europe: 2003 water-table fluctuations represented by 3 features



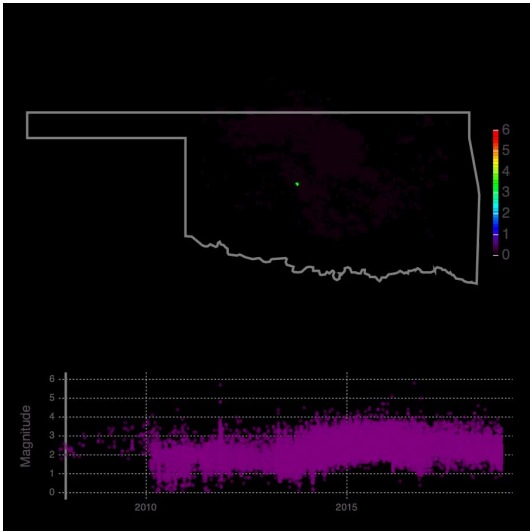
Climate model of Europe: analyze all model outputs (>40) simultaneously



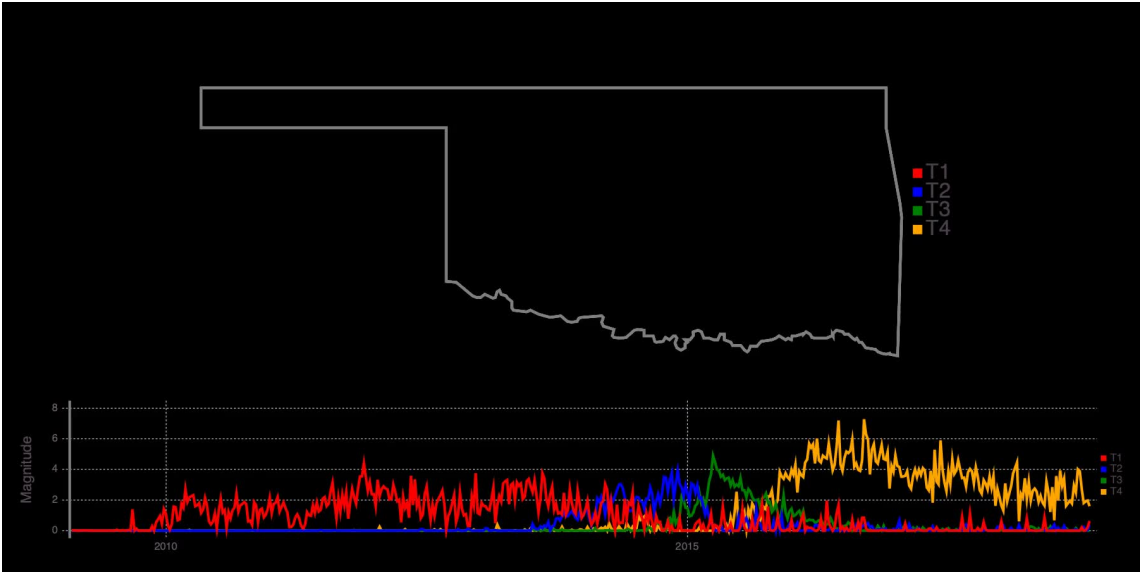
- Find interconnections among model outputs
- Evaluate impacts of different model setups
- Find dominant processes impacting model predictions
  (e.g., climate impacts on groundwater resources, impacts of subsurface processes on atmospheric conditions)

Oklahoma seismic events 1991-2018

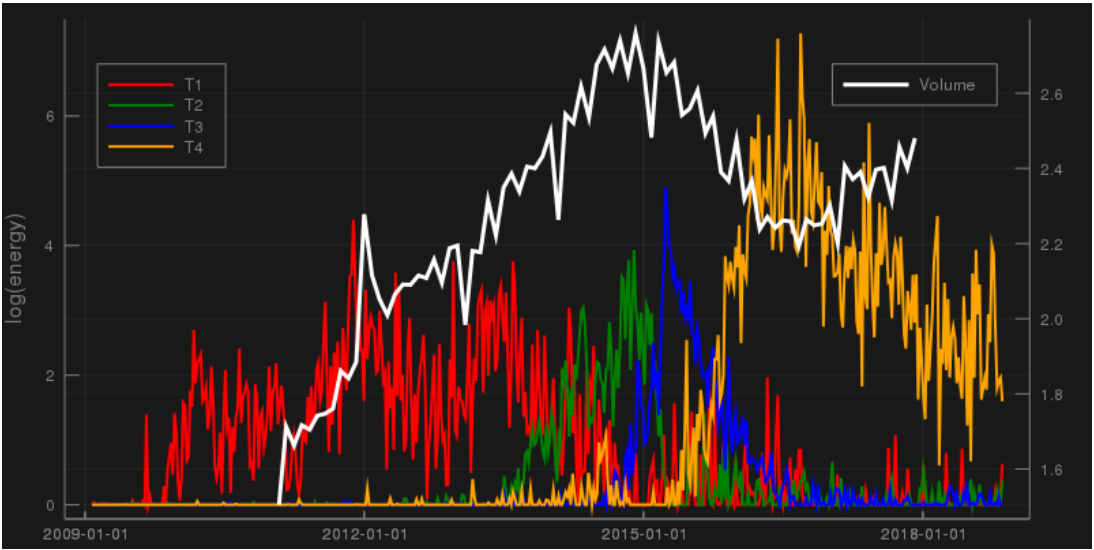
- 32,251 seismic events from 1989 to 2017
- Tensor: total energy of events over a discretized domain
  $(118 \times 97 \times 520)$
  $(columns \times rows \times weeks)$
- **NTF**k applied to extract dominant hidden (latent) features based on spatial footprints and temporal characteristics



Oklahoma seismic events 1991-2018: reconstruction by 4 features (signals)

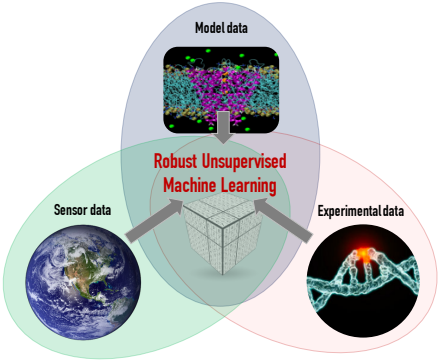


Oklahoma seismic events 1991-2018: extracted signals vs. injected volumes



Summary

- Developed novel unsupervised ML methods and computational tools based on Nonnegative Factorization (Matrices/Tensors)
- Our ML methods have been used to solve various real-world problems
- Our goal is to further tests our algorithms on diverse datasets

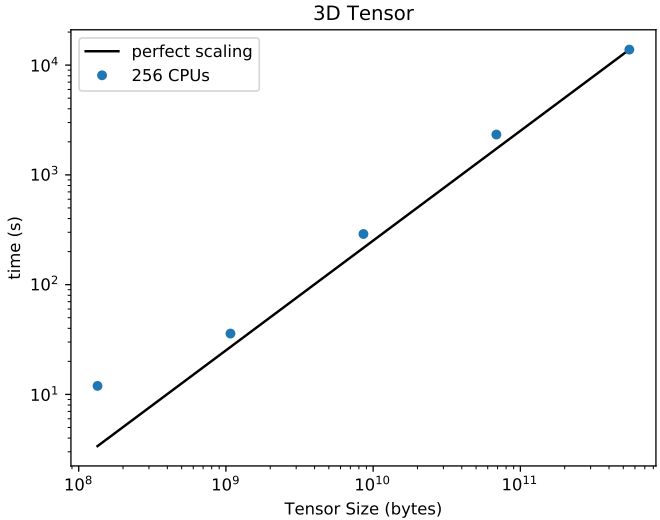


- **NMFk** + ShiftNMFk + GreenNMFk
- **NTFk**
- **NBMF**: Quantum machine learning using **D-Wave** quantum annealer
- **MADS**: Model-Analyses & Decision Support
  `http://mads.gitlab.io` `http://madsjulia.github.io/Mads.jl`
- Feature extraction examples:
  `http://madsjulia.github.io/Mads.jl/Examples/blind_source_separation`
- Slide deck / publications: `http://monty.gitlab.io`

- **Identifying the number of unknown features**:
  - applying custom k-means clustering and sparsity constraints
  - analyzing reconstruction quality (e.g., Frobenius norm) and cluster Silhouettes
- **Solving a non-unique optimization problem**:
  - performing multistarts, regularization and nonnegativity constraints
  - applying diverse optimization techniques (Multiplicative/Alternating Least Squares algorithms, NLopt, Ipopt, Gurobi, MOSEK, GLPK, Clp, Cbc, ...)
  - accounting for the physics
- **Processing Big Data**:
  - GPU's / TPU's / Distributed computing
  - Account for data sparsity and structure
  - Nonnegative Tensor Trains
- **Dealing with Noisy Data**:
  - Random noise impacts accuracy but its accountable
  - Systematic noise is identified as separate signals (features)

4GB Tensor ($1000 \times 1000 \times 1000$)

Framework	Execution time (seconds)
MATLAB	2634
NumPy	881
MXNet	644
PyTorch	121
TensorFlow	119
Julia	109


3D Tensor
perfect scaling
256 CPUs
time (s)
10^4
10^3
10^2
10^1
10^8
10^9
10^{10}
10^{11}
Tensor Size (bytes)

- **Data Analytics**: Identify signals (features) in datasets (latent variables)
  - Feature extraction (**FE**):
  - Blind source separation (**BSS**)
  - Detection of disruptions / anomalies
  - Image recognition
  - Discover unknown dependencies and phenomena
  - Guide development of physics / reduced-order models representing the data

- **Model Analytics/Diagnostics**: Identify processes (features) in model outputs
  - Identify dependencies between model inputs and outputs
  - Discover unknown dependencies
  - Separate processes (inseparable during modeling)
  - Develop ML (reduced-order) models

- **Coupled Data/Model Analytics**:
  Simultaneous analyses of data and model outputs (data/model fusion)