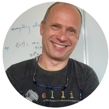
From biased to unbiased dynamics: slow dynamical modes from static averages

September 16, 2025

Timothée Devergne

Vladimir Kostic


Massimiliano Pontil

Michele Parrinello

ISTITUTO ITALIANO DI TECNOLOGIA

Funded by the European Union NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA

FAIR Future Artificial Intelligence Research

Outline

1 Overview: what is our aim here?

Outline

1. Overview: what is our aim here?
2. Dynamical quantities from unbiased simulations

Outline

1. Overview: what is our aim here?
2. Dynamical quantities from unbiased simulations
3. How can we transfer these concepts to biased simulations?

Outline

1. Overview: what is our aim here?
2. Dynamical quantities from unbiased simulations
3. How can we transfer these concepts to biased simulations?
4. Results on models and application to real life intrisically disordered proteins

Outline

1. Overview: what is our aim here?
2. Dynamical quantities from unbiased simulations
3. How can we transfer these concepts to biased simulations?
4. Results on models and application to real life intrisically disordered proteins
5. Blind discovery of metastable states

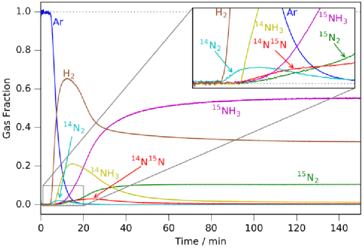
Studying the dynamics

We are interested in transitions between states and their evolution over time: at time t, in which state will the system be?

Studying the dynamics

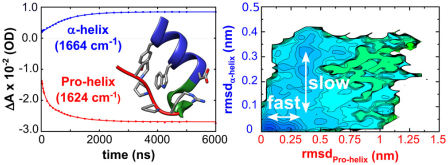
We are interested in transitions between states and their evolution over time: at time t, in which state will the system be?

Experimental characterization of a chemical reaction



Wood *et al.*, **Phys. Chem. Chem. Phys.**, 2017, 19, 4719-4724

Experimental/theoretical study of protein folding



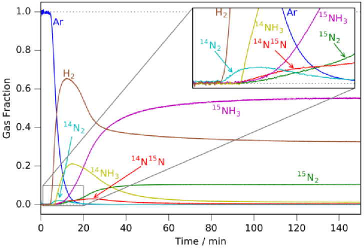
Meuzelaar *et al.*, **J. Phys. Chem. B**, 2013, 117, 39, 11490–11501

Studying the dynamics

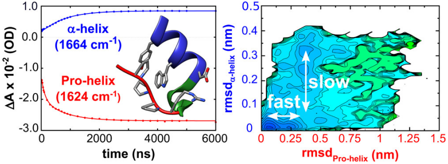
We are interested in transitions between states and their evolution over time: at time *t*, in which state will the system be?

Experimental characterization of a chemical reaction

Experimental/theoretical study of protein folding



Wood *et al.*, **Phys. Chem. Chem. Phys.**, 2017, 19, 4719-4724



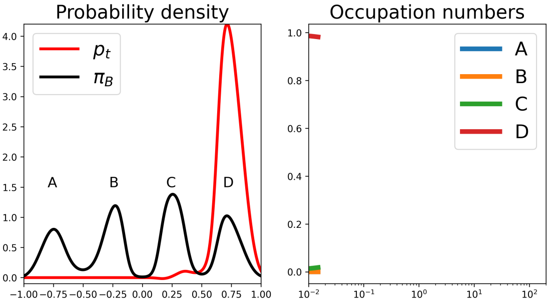
Meuzelaar *et al.*, **J. Phys. Chem. B**, 2013, 117, 39, 11490–11501

A probabilistic view on using data from molecular dynamics to study time evolution of systems

Occupation numbers are related to probability distributions

Occupation number of state A at time t $\mathbb{1}_A(t)$: $\mathbb{1}_{\mathbb{A}}(t) = \int_{x \in A} p_t(x)dx$

p_t: Probability distribution at time t, converges towards the Boltzmann one for $t \to \infty$

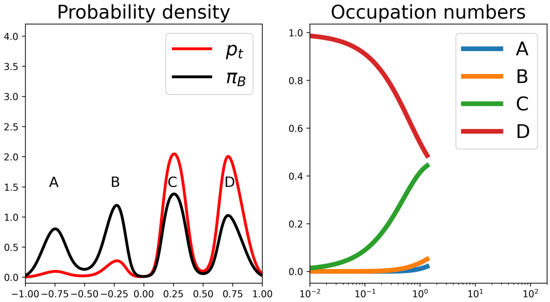

Probability density
Occupation numbers

- Knowing p_t gives access to the time evolution of any observable
- Starting in a given state, how will the occupation number evolve with time?

Occupation numbers are related to probability distributions

Occupation number of state A at time t $\mathbb{1}_A(t)$: $\mathbb{1}_{\mathbb{A}}(t) = \int_{x \in A} p_t(x)dx$

p_t: Probability distribution at time t, converges towards the Boltzmann one for $t \to \infty$


Probability density

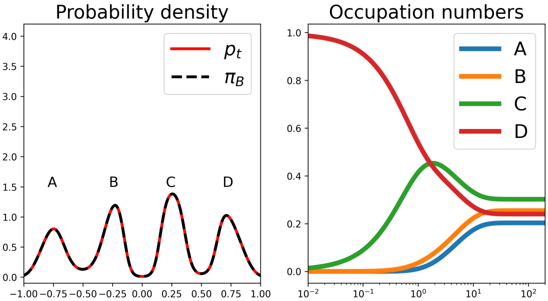
Occupation numbers

- Knowing p_t gives access to the time evolution of any observable
- Starting in a given state, how will the occupation number evolve with time?

Occupation numbers are related to probability distributions

Occupation number of state A at time t $\mathbb{1}_A(t)$: $\mathbb{1}_{\mathbb{A}}(t) = \int_{x \in A} p_t(x)dx$

p_t: Probability distribution at time t, converges towards the Boltzmann one for $t \to \infty$



Probability density

Occupation numbers

- Knowing p_t gives access to the time evolution of any observable
- Starting in a given state, how will the occupation number evolve with time?
- This is for a very simple system, how can we have p_t for multidimensional systems

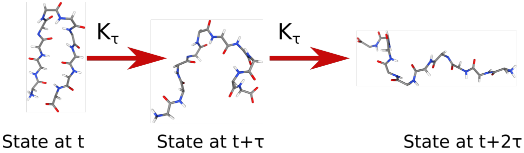
The Koopman/Transfer Operator from Unbiased Simulations

Given a Markov process, define

$$u_t = \frac{p_t}{\pi},$$

then the Koopman operator K_τ propagates u_t forward in time:

$$u_{t+\tau} = K_\tau u_t \quad \Longrightarrow \quad u_{n\tau} = (K_\tau)^n u_0.$$



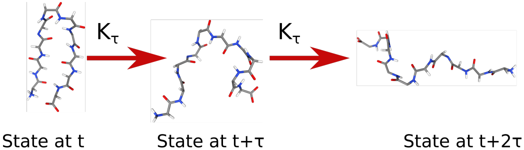
The Koopman/Transfer Operator from Unbiased Simulations

Given a Markov process, define

$$u_t = \frac{p_t}{\pi},$$

then the Koopman operator K_τ propagates u_t forward in time:

$$u_{t+\tau} = K_\tau u_t \quad \Longrightarrow \quad u_{n\tau} = (K_\tau)^n u_0.$$



Learning K_τ: Several approaches exist (EDMD, kernel methods, VAMP, etc.):

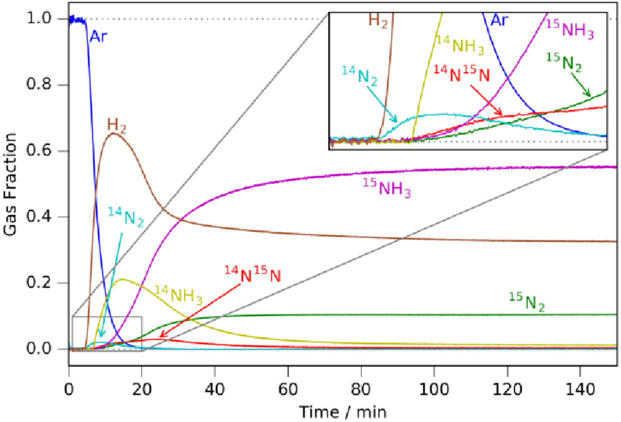
- Schütte et al., Springer, 1999
- Mardt *et al.*, Nature Communications, 2018
- Kostic *et al.*, ICLR 2024

All require: The evaluation of the time-lagged correlation function

$$C(\tau) = \int \psi(x_t)\,\psi(x_{t+\tau})\,dt,$$

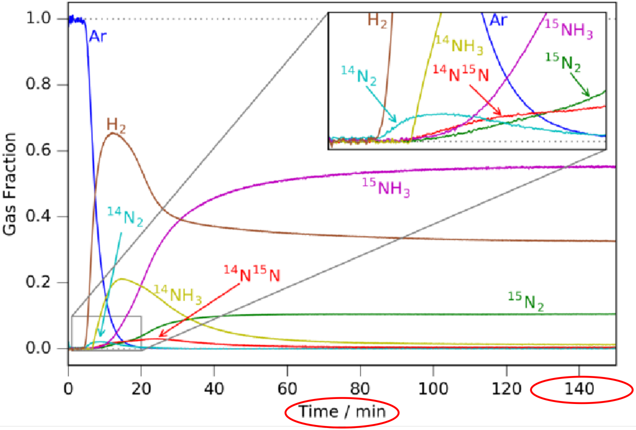
where ψ is a chosen observable.

Time evolution of reactions



- Transition mechanisms often involve several steps

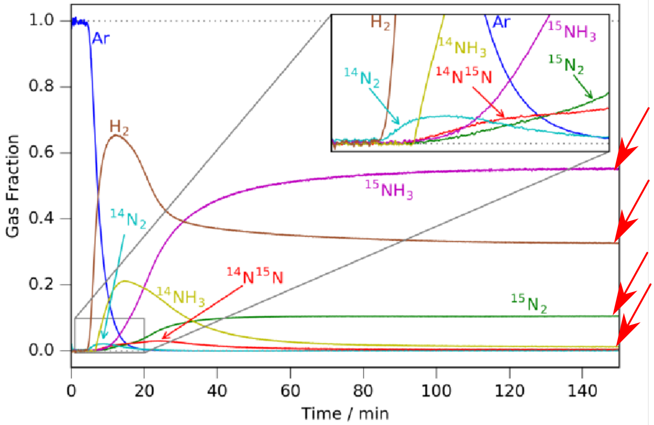
Time evolution of reactions



Wood *et al.*, **Phys. Chem. Chem. Phys.**, 2017, 19, 4719-4724

- Transition mechanisms often involve several steps
- They are rare events
- The typical timestep of a simulation : $0.2 \rightarrow 2fs$
- 100 minutes would mean $\approx 6.10^{18}$ timesteps which is not possible
- Some form of enhanced sampling scheme is needed

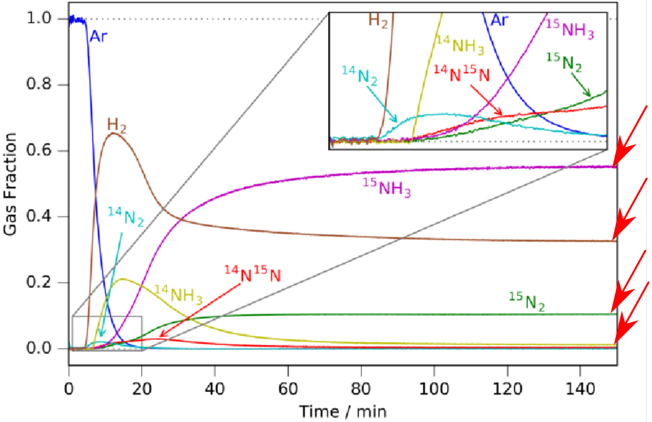
What observable can we get from a single long biased simulations



- The free energy profile gives occupation numbers at equilibrium

Wood *et al.*, **Phys. Chem. Chem. Phys.**, 2017, 19, 4719-4724

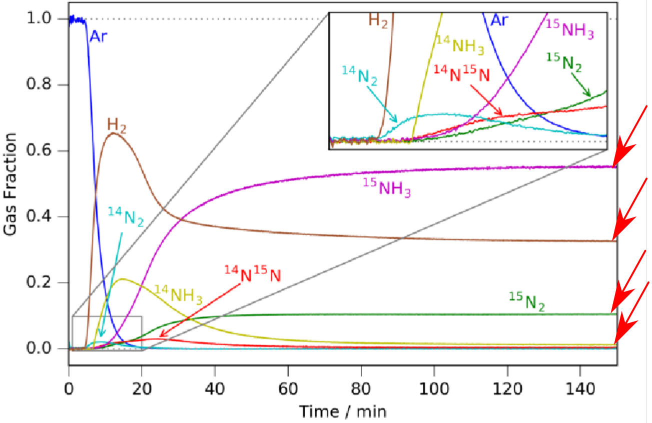
What observable can we get from a single long biased simulations



- The free energy profile gives occupation numbers at equilibrium
- An estimate of the rates can be obtained by the free energy barrier

Wood *et al.*, **Phys. Chem. Chem. Phys.**, 2017, 19, 4719-4724

What observable can we get from a single long biased simulations



Wood *et al.*, **Phys. Chem. Chem. Phys.**, 2017, 19, 4719-4724

- The free energy profile gives occupation numbers at equilibrium
- An estimate of the rates can be obtained by the free energy barrier
- How can we get the time evolution of the occupation numbers of the states?
- $C_{biased}(\tau) \neq C_{unbiased}(\tau)$

Is there a differential equation for u_t?

From discrete to continuous time

$$u_{t+\tau} = K_\tau \, u_t$$

The difficulty: we need u at both t and $t+\tau$. What happens as $\tau \to 0 : \frac{\partial u_t}{\partial t}$?

Is there a differential equation for u_t?

From discrete to continuous time

$$u_{t+\tau} = K_\tau \, u_t$$

The difficulty: we need u at both t and $t + \tau$. What happens as $\tau \to 0$: $\frac{\partial u_t}{\partial t}$?

Langevin dynamics assumption

There is a closed-form equation: $\frac{\partial u_t}{\partial t} = \mathcal{L} u_t$

$\mathcal{L}$ is the **infinitesimal generator** of the dynamics (backward Kolmogorov equation):

$$\mathcal{L} f(\mathrm{R}) = \frac{1}{\gamma} \sum_{i=1}^{N} \frac{1}{m_i} \frac{\partial f(\mathrm{R})}{\partial r_i} \frac{\partial U(\mathrm{R})}{\partial r_i} - \frac{1}{\beta\gamma} \sum_{i=1}^{N} \frac{1}{m_i} \frac{\partial^2 f(\mathrm{R})}{\partial r_i^2}$$

Is there a differential equation for u_t?

From discrete to continuous time

$$u_{t+\tau} = K_\tau \, u_t$$

The difficulty: we need u at both t and $t+\tau$. What happens as $\tau \to 0$: $\frac{\partial u_t}{\partial t}$?

Langevin dynamics assumption

There is a closed-form equation: $\frac{\partial u_t}{\partial t} = \mathcal{L} u_t$

$\mathcal{L}$ is the **infinitesimal generator** of the dynamics (backward Kolmogorov equation):

$$\mathcal{L} f(\mathrm{R}) = \frac{1}{\gamma} \sum_{i=1}^{N} \frac{1}{m_i} \frac{\partial f(\mathrm{R})}{\partial r_i} \frac{\partial U(\mathrm{R})}{\partial r_i} - \frac{1}{\beta\gamma} \sum_{i=1}^{N} \frac{1}{m_i} \frac{\partial^2 f(\mathrm{R})}{\partial r_i^2}$$

- $\mathcal{L}$ depends linearly on the potential U
- But this formulation is cumbersome and only tractable in low dimensions

Modal decomposition of $\mathcal{L}$

$\mathcal{L}$ can be decomposed into eigenvectors and eigenvalues which give insights about the dynamical behavior:

$$\mathcal{L}\psi_i(\mathrm{R}) = \lambda_i\psi_i(\mathrm{R}), \quad 0 = \lambda_0 < \lambda_1 < \lambda_2 < \dots$$

$$u_t(\mathrm{R}) = \sum_i \psi_i(\mathrm{R})e^{-\lambda_i t} \langle\psi_i|u_0\rangle$$

$\langle\psi_i|u_0\rangle = \int \psi_i(\mathrm{R})u_0(\mathrm{R})\frac{e^{-\beta U(\mathrm{R})}}{Z}d\mathrm{R}$

In other words, if you sampled the Boltzmann distribution, you have everything

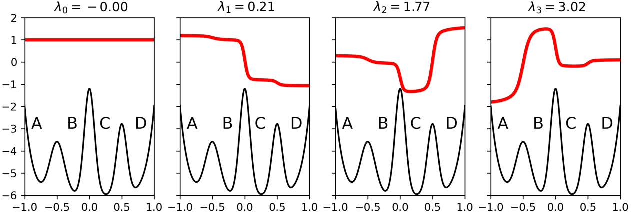
Modal decomposition of $\mathcal{L}$

$\mathcal{L}$ can be decomposed into eigenvectors and eigenvalues which give insights about the dynamical behavior:

$$\mathcal{L}\psi_i(\mathrm{R}) = \lambda_i\psi_i(\mathrm{R}), \quad 0 = \lambda_0 < \lambda_1 < \lambda_2 < \dots$$

$$u_t(\mathrm{R}) = \sum_i \psi_i(\mathrm{R})e^{-\lambda_i t}\,\langle\psi_i|u_0\rangle$$

$\langle\psi_i|u_0\rangle = \int \psi_i(\mathrm{R})u_0(\mathrm{R})\frac{e^{-\beta U(\mathrm{R})}}{Z}d\mathrm{R}$
In other words, if you sampled the Boltzmann distribution, you have everything



The problem is transformed from a PDE to an eigenvalue problem

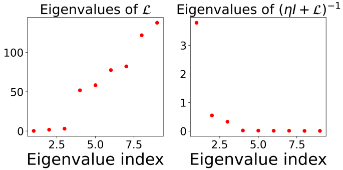
The machine learning problem:

Given a (biased) trajectory, how can we learn the eigenpairs of the system?

The machine learning problem:

Given a (biased) trajectory, how can we learn the eigenpairs of the system?

Interesting eigenvalues are close to zero, while the others grow fast:

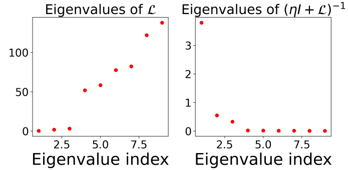


Instead of $\mathcal{L}$, we will learn $(\eta I + \mathcal{L})^{-1}$ which has the same eigenfunctions as $\mathcal{L}$

The machine learning problem:

Given a (biased) trajectory, how can we learn the eigenpairs of the system?

Interesting eigenvalues are close to zero, while the others grow fast:



Instead of $\mathcal{L}$, we will learn $(\eta I + \mathcal{L})^{-1}$ which has the same eigenfunctions as $\mathcal{L}$

$\mathcal{L}$ involves second derivatives

We will use the Dirichlet form:

$$\langle\psi|\mathcal{L}|\phi\rangle = -\frac{1}{\beta\gamma}\int \nabla_{\mathbf{u}}\phi(\mathbf{R})\nabla_{\mathbf{u}}\psi(\mathbf{R})\frac{e^{-\beta U(\mathbf{R})}}{Z}d\mathbf{R}$$

Dirichlet

Kolmogorov

Operator learning

Given a set of functions $|\phi_i\rangle$, one will try to find the matrix G such that

$$(\eta I + \mathcal{L})^{-1}\phi_i(x) \approx \sum_j G_{ij}\phi_j(x)$$

Operator learning

Given a set of functions $|\phi_i\rangle$, one will try to find the matrix G such that

$$(\eta I + \mathcal{L})^{-1}\phi_i(x) \approx \sum_j G_{ij}\phi_j(x)$$

This problem has a closed form

$$G = W^{-1}S \text{ where } W_{ij} = \langle\phi_i|\eta I + \mathcal{L}|\phi_j\rangle \text{ and } S_{ij} = \langle\phi_i|\phi_j\rangle$$

Operator learning

Given a set of functions $|\phi_i\rangle$, one will try to find the matrix G such that

$$(\eta I + \mathcal{L})^{-1}\phi_i(x) \approx \sum_j G_{ij}\phi_j(x)$$

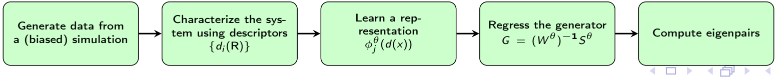
This problem has a closed form

$$G = W^{-1}S \text{ where } W_{ij} = \langle\phi_i|\eta I + \mathcal{L}|\phi_j\rangle \text{ and } S_{ij} = \langle\phi_i|\phi_j\rangle$$

The $|\phi_i\rangle$ are learned by using neural networks, minimizing the loss $Tr(S^\theta\Lambda^\theta W^\theta\Lambda^\theta - 2S^\theta\Lambda^\theta) + Tr((S^\theta - I)^2)$

Boltzmann distribution is all we need

There is no time dependance, it is replaced with the differential operator, only Boltzmann averages are needed.



Generate data from a (biased) simulation

Characterize the system using descriptors $\{d_i(\mathrm{R})\}$

Learn a representation $\phi_j^\theta(d(x))$

Regress the generator $G = (W^\theta)^{-\mathbf{1}}S^\theta$

Compute eigenpairs

Alanine dipeptide: first eigenfunction as CV

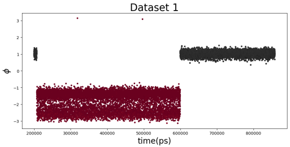
Test of this method on a model system: alanine dipeptide

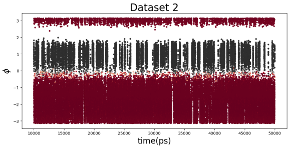
- ϕ is known to be good collective variable: biasing with it will lead to many transitions
- ψ is known to be a bad collective variable

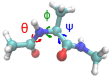
Alanine dipeptide: first eigenfunction as CV

Test of this method on a model system: alanine dipeptide

- ϕ is known to be good collective variable: biasing with it will lead to many transitions
- ψ is known to be a bad collective variable
- We trained the models on two trajectories: one biased on ψ, the other one ϕ
- The (ϕ, θ) plane is known to represent well the transition state Proc. Natl. Acad. Sci. U.S.A. 97 (11) 5877-5882



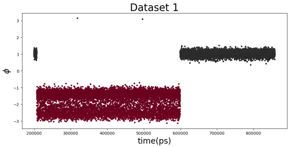


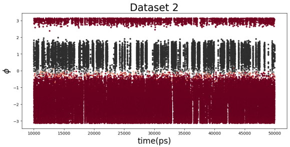


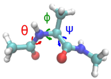
Alanine dipeptide: first eigenfunction as CV

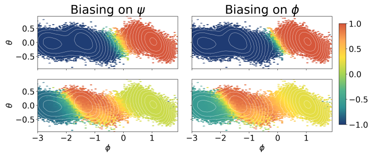
Test of this method on a model system: alanine dipeptide

- ϕ is known to be good collective variable: biasing with it will lead to many transitions
- ψ is known to be a bad collective variable
- We trained the models on two trajectories: one biased on ψ, the other one ϕ
- The (ϕ, θ) plane is known to represent well the transition state Proc. Natl. Acad. Sci. U.S.A. 97 (11) 5877-5882









Conclusion

We can obtain a good estimate of where the transition region is, even with a sparse dataset

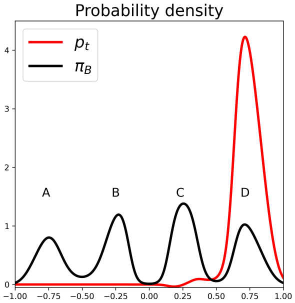
Time evolution of observables

We start from a system in a metastable state A and we want to know its relaxation towards the Boltzmann distribution:

The initial distribution is the Boltzmann one restricted to state D:

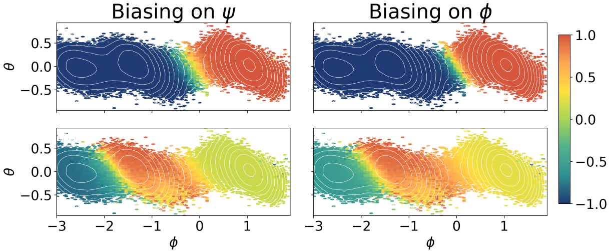
$$p_0^D(\mathrm{R}) = \frac{e^{-\beta U(\mathrm{R})}}{\int_{\mathrm{R}\in D} e^{-\beta U(\mathrm{R})}} \mathbb{1}_D(\mathrm{R}) \text{ hence } u_0^D(\mathrm{R}) = \frac{Z}{Z_A} \mathbb{1}_D(\mathrm{R})$$

$$\mathrm{u}_t(\mathrm{R}) = \sum_{i=0}^m \psi_i(\mathrm{R}) e^{-\lambda_i t} \langle \psi_i | u_0 \rangle$$



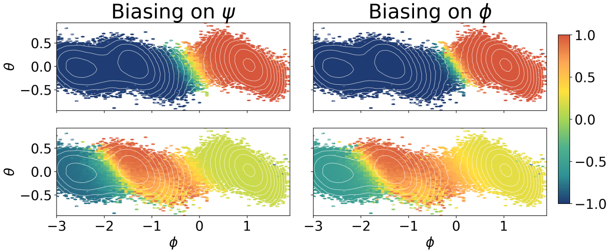
Time evolution of alanine dipeptide

The sign of eigenfunctions can be used to classify states due to the relation of $\int \psi_i(\mathrm{R}) \frac{e^{-\beta U(\mathrm{R})}}{Z} d\mathrm{R} = 0$

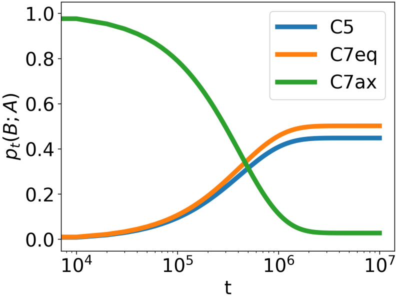


Time evolution of alanine dipeptide

The sign of eigenfunctions can be used to classify states due to the relation of $\int \psi_i(\mathrm{R}) \frac{e^{-\beta U(\mathrm{R})}}{Z} d\mathrm{R} = 0$

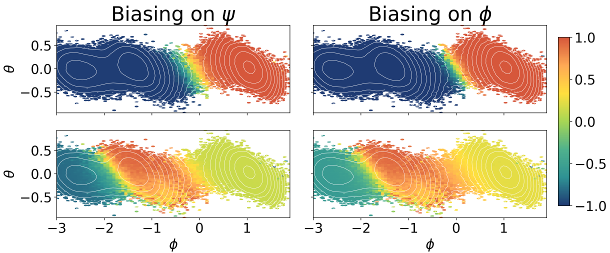


And forecast time evolution of occupation numbers

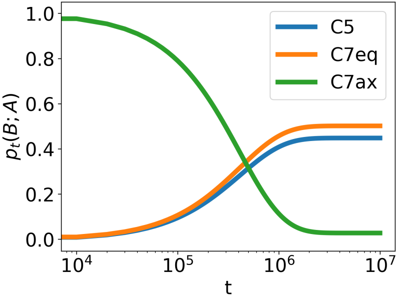


Time evolution of alanine dipeptide

The sign of eigenfunctions can be used to classify states due to the relation of $\int \psi_i(\mathrm{R})\dfrac{e^{-\beta U(\mathrm{R})}}{Z}d\mathrm{R} = 0$



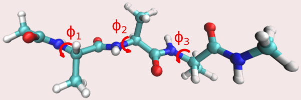
And forecast time evolution of occupation numbers



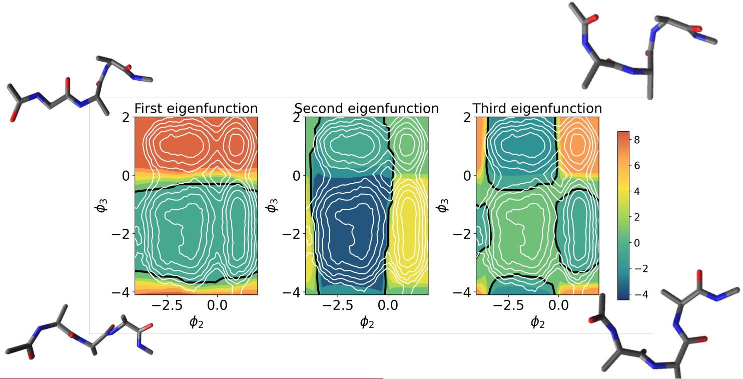
From an initial propability distribution in the least occupied state, we can forecast how the other state will evolve. Can we do it for systems with a higher number of states?

Alanine tetrapeptide

Alanine tetrapeptide is a simple molecule displaying several states

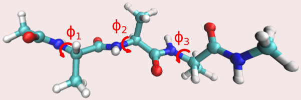


Identification of the states

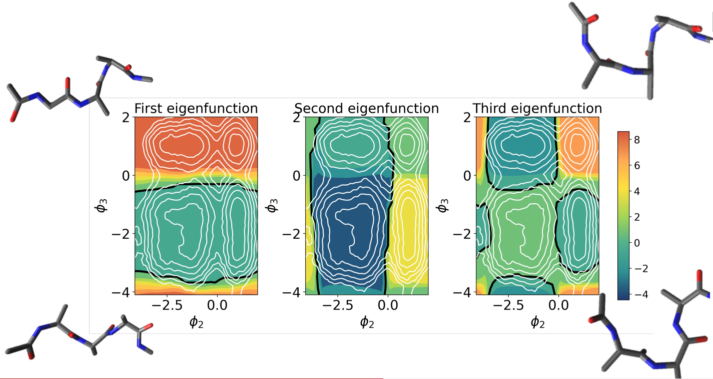


Alanine tetrapeptide

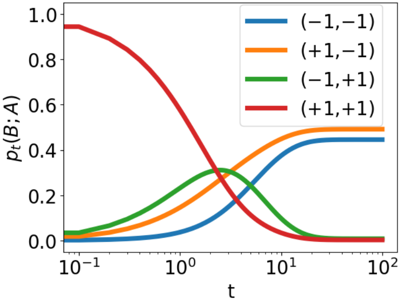
Alanine tetrapeptide is a simple molecule displaying several states



Identification of the states



Forecasting occupation numbers

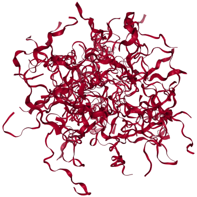


Until there: only small molecules for which we already know the results.

Application: Intrinsically disordered proteins.

IDPs are proteins that do not have a fixed stable 3D structure.

- Disorder is present in 70% of proteins
- They are involved in many cancers and degenerative diseases like Alzheimer or Parkinson
- Considered undrugable due to the high number of conformations they can adopt
- ML models (alphafold..) describe them poorly



Peptone

M. Invernizzi and S. Bottaro

Application: IDPs Aβ42

Aβ42 is a 42 residues IDP involved in Alzheimer disease: finding its most stable states is key. For this study, we used a publicly available (unbiased) trajectory Lohr *et al* Nature Computational Science (2021).

Application: IDPs Aβ42

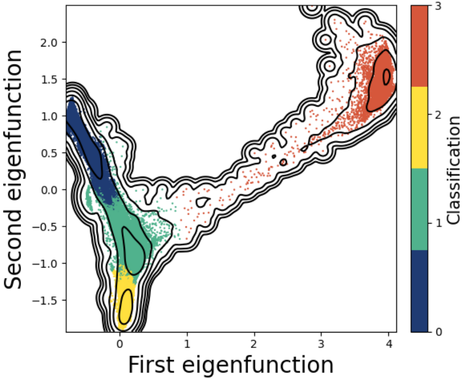
Aβ42 is a 42 residues IDP involved in Alzheimer disease: finding its most stable states is key. For this study, we used a publicly available (unbiased) trajectory Lohr *et al* Nature Computational Science (2021).

- The eigenfunctions give us a physical low dimensional space
- Clustering using gaussian mixture model gives great results

Application: IDPs Aβ42

Aβ42 is a 42 residues IDP involved in Alzheimer disease: finding its most stable states is key. For this study, we used a publicly available (unbiased) trajectory Lohr *et al* Nature Computational Science (2021).

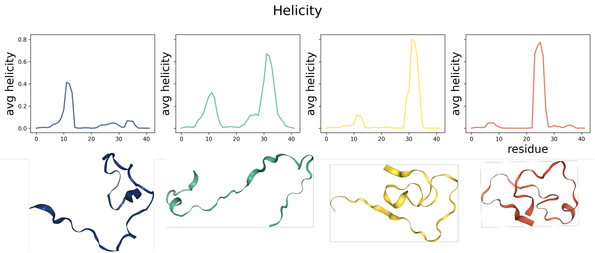
- The eigenfunctions give us a physical low dimensional space
- Clustering using gaussian mixture model gives great results

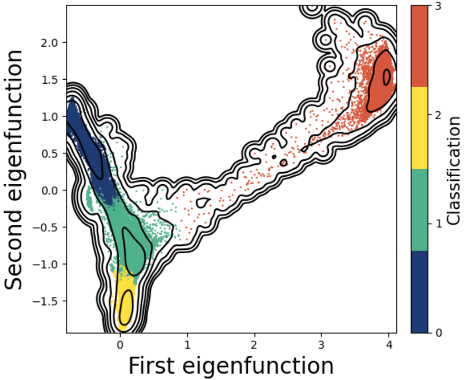


Are these states physically meaningful ? Let us look at the secondary structures

Application: IDPs Aβ42 secondary structure

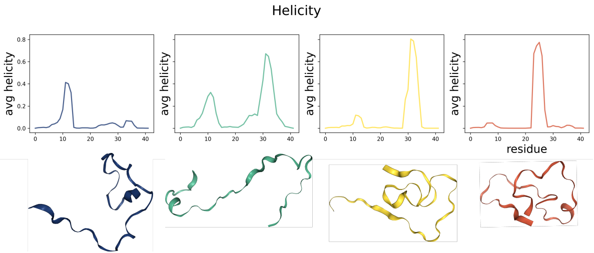
All states are related to a secondary structure change.

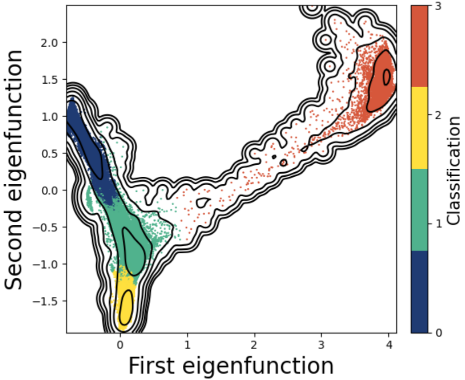




Application: IDPs Aβ42 secondary structure

All states are related to a secondary structure change.





We can associate each mode with a structural change, **because the eigenfunctions contain the relevant long time dynamical behavior information**

Conclusion I

Main idea

- Compute dynamical properties directly from Boltzmann-distributed samples
- In principle, applicable with *any* data generation scheme

Conclusion I

Main idea

- Compute dynamical properties directly from Boltzmann-distributed samples
- In principle, applicable with *any* data generation scheme

Ongoing work

- Use of eigenfunctions as collective variables for fast-folding proteins (**see J. Yang's poster, Thursday**)

Conclusion I

Main idea

- Compute dynamical properties directly from Boltzmann-distributed samples
- In principle, applicable with *any* data generation scheme

Ongoing work

- Use of eigenfunctions as collective variables for fast-folding proteins (**see J. Yang's poster, Thursday**)

Open question

$$\mathcal{L} = \mathrm{Tr}\big(S^\theta \Lambda^\theta W^\theta \Lambda^\theta - 2S^\theta \Lambda^\theta\big) + \mathrm{Tr}\big((S^\theta - I)^2\big)$$

Where are the states? **Nowhere.**

Conclusion I

Main idea

- Compute dynamical properties directly from Boltzmann-distributed samples
- In principle, applicable with *any* data generation scheme

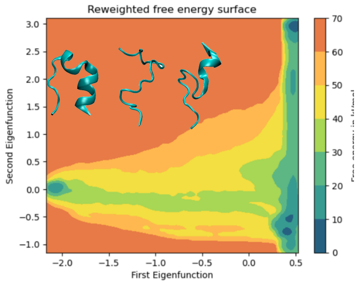
Ongoing work

- Use of eigenfunctions as collective variables for fast-folding proteins (**see J. Yang's poster, Thursday**)

Open question

$$\mathcal{L} = \mathrm{Tr}\left(S^\theta \Lambda^\theta W^\theta \Lambda^\theta - 2S^\theta \Lambda^\theta\right) + \mathrm{Tr}\left((S^\theta - I)^2\right)$$

Where are the states? **Nowhere.**

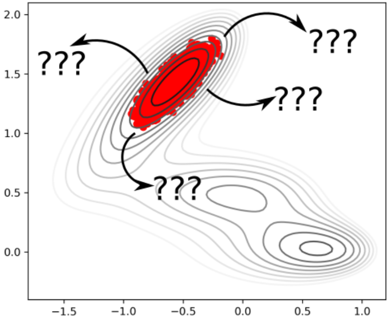


Can we start from only one state?

The loss

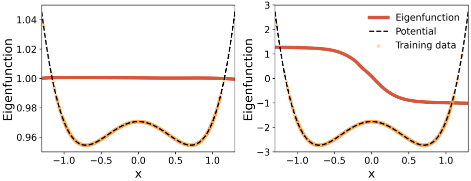
$$\mathcal{L} = \mathrm{Tr}\big(S^\theta \Lambda^\theta W^\theta \Lambda^\theta - 2S^\theta \Lambda^\theta\big) + \mathrm{Tr}\big((S^\theta - I)^2\big)$$

Is this only a postprocessing tool?



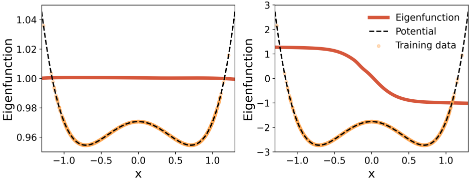
Our initial observation

At high temperature, when the whole landscape is sampled:

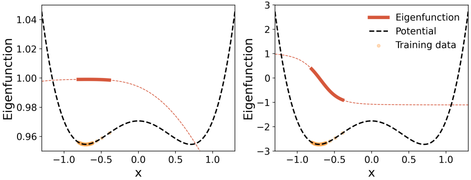


Our initial observation

At high temperature, when the whole landscape is sampled:

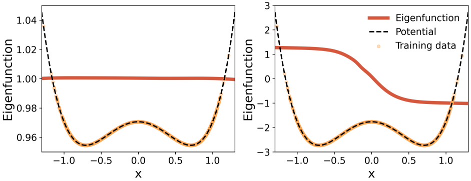


At low temperature, when only one state is sampled

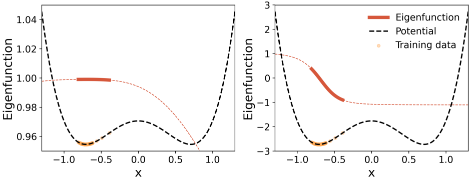


Our initial observation

At high temperature, when the whole landscape is sampled:



At low temperature, when only one state is sampled



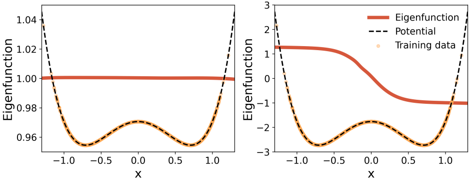
We use what is thought to be a pain for neural networks: their poor extrapolation capabilities

Use the curvature that defines the unexplored space:

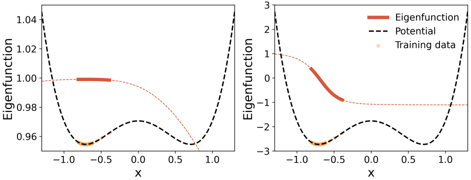
$$V_K(x) = -\frac{\lambda}{\beta} \ln\left(|\nabla\psi_0(x)|^2 + \epsilon\right)$$

Our initial observation

At high temperature, when the whole landscape is sampled:



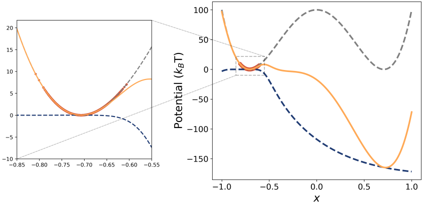
At low temperature, when only one state is sampled



We use what is thought to be a pain for neural networks: their poor extrapolation capabilities

Use the curvature that defines the unexplored space:

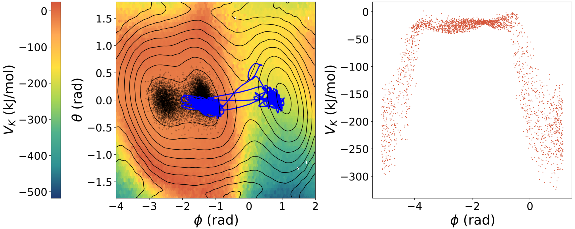
$$V_K(x) = -\frac{\lambda}{\beta}\ln\left(|\nabla\psi_0(x)|^2+\epsilon\right)$$



$V_K(x)$ is called the Kolmogorov bias (Kang *et al*, Nature Computational Science, 2024).

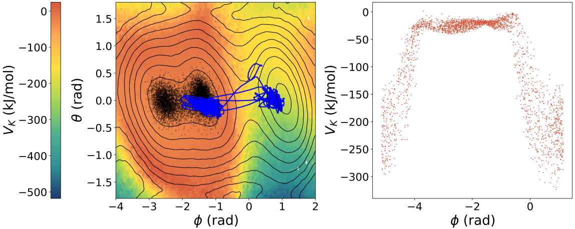
Proof of concept: alanine dipeptide

We mix V_K with an OPES bias to push the system out of the initial basin



Proof of concept: alanine dipeptide

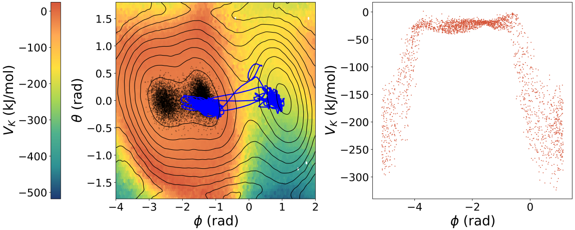
We mix V_K with an OPES bias to push the system out of the initial basin



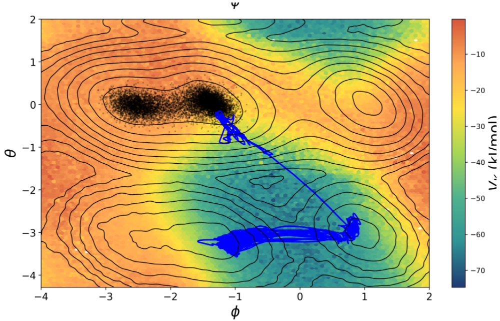
We can also discover new states

Proof of concept: alanine dipeptide

We mix V_K with an OPES bias to push the system out of the initial basin

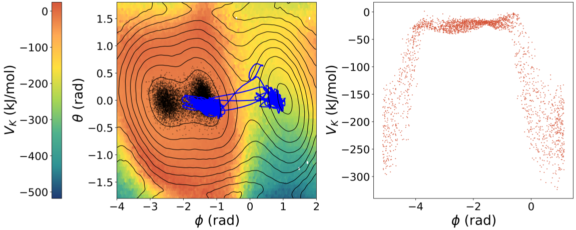


We can also discover new states

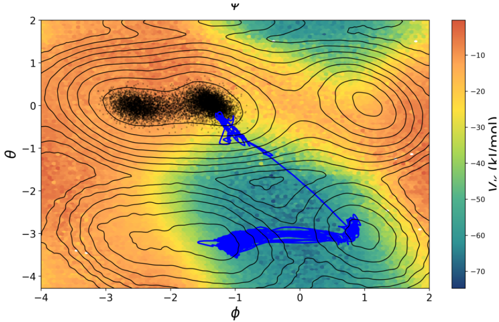


Proof of concept: alanine dipeptide

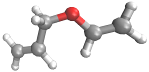
We mix V_K with an OPES bias to push the system out of the initial basin



We can also discover new states

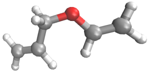


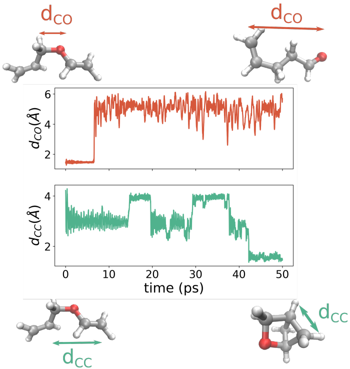
Discovering chemical reactions

Given an initial reactant, how to predict the possible products 

Discovering chemical reactions

Given an initial reactant, how to predict the possible products

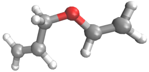


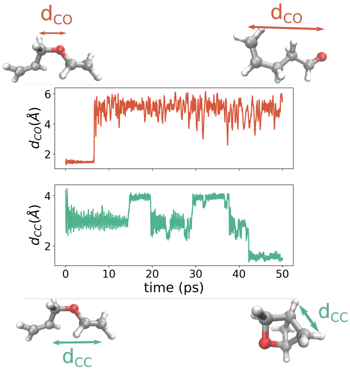


We obtained species already reported in the literature: Raucci *et al.*, J. Phys. Chem. Lett., 2022

Discovering chemical reactions

Given an initial reactant, how to predict the possible products





We obtained species already reported in the literature: Raucci *et al.*, J. Phys. Chem. Lett., 2022

Possible application

Prebiotic chemistry: how did bioprecursors form on earth

Conclusions

Take-home messages

- We can extract dynamical information directly from Boltzmann averages Devergne *et al* J. Phys. Chem, 2025

Conclusions

Take-home messages

- We can extract dynamical information directly from Boltzmann averages Devergne *et al* J. Phys. Chem, 2025
- Eigenfunctions provide a low-dimensional space to analyse the data

Conclusions

Take-home messages

- We can extract dynamical information directly from Boltzmann averages Devergne *et al* J. Phys. Chem, 2025
- Eigenfunctions provide a low-dimensional space to analyse the data
- New states can be discovered even from an unbiased simulation in a metastable basin Devergne *et al.*, arXiv:2508.01477

Conclusions

Take-home messages

- We can extract dynamical information directly from Boltzmann averages Devergne *et al* J. Phys. Chem, 2025
- Eigenfunctions provide a low-dimensional space to analyse the data
- New states can be discovered even from an unbiased simulation in a metastable basin Devergne *et al.*, arXiv:2508.01477

mlcolvar

Implementation in mlcolvar
Thanks to Enrico Trizio

Code snippet

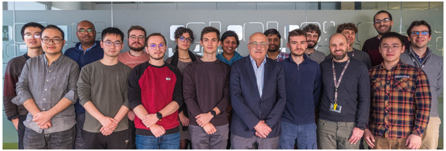
```
smart_derivatives = SmartDerivatives(force_all_atoms=True)
smart_dataset = smart_derivatives.setup(
    dataset, ComputeDistances, n_atoms,
    descriptors_batch_size=1000, positions_noise=1e-4)

model = Generator(
    layers=[45,20,20,1], eta=0.005, r=3, alpha=0.005,
    friction=friction, descriptors_derivatives=smart_derivatives,
    options=options, u_stat=True, cell=cell)
```

Joyeux anniversaire Michele

"Grazie per la tua pazienza"

Thank you



Atomistic simulations@IIT

CSML@IIT