



Inria



sanofi

European Research Council  
Established by the European Commission

# Coarse-graining and efficiently sampling with autoencoders

**Gabriel STOLTZ**

(CERMICS, Ecole des Ponts & MATHEMATICALS team, Inria Paris)

*With Z. Belkacemi (Sanofi & ENPC), T. Lelièvre (ENPC/Inria) and P. Gkeka (Sanofi)*

MIF++ seminar, February 3rd, 2023

- **A (short/biased) review of machine learning approaches for CV**
- **Free-energy biasing and iterative learning with autoencoders<sup>1</sup>**
  - **Autoencoders**: definition, training, interpretation
  - **Extended** adaptive biasing force method
  - General presentation of the iterative algorithm
  - Illustration/sanity checks on toy examples
- **Applications to systems of interest** (alanine dipeptide, chignolin, HSP90)

---

<sup>1</sup>Z. Belkacemi, P. Gkeka, T. Lelièvre, G. Stoltz, *J. Chem. Theory Comput.* **18** (2022)

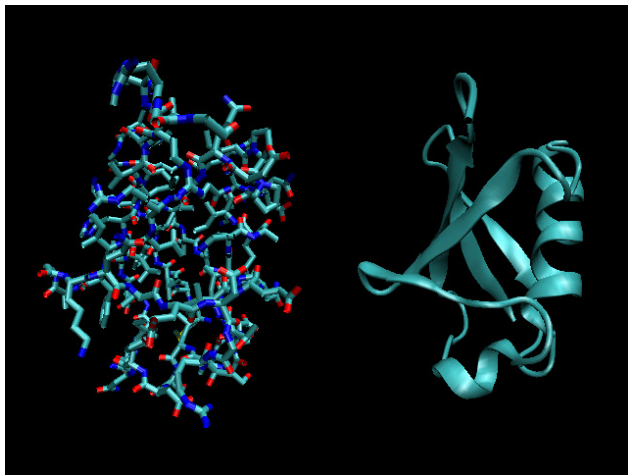
# (A biased perspective on some) References

- **ML reviews in MD** (biased towards dimensionality reduction, not force fields)
  - A. Gliemlo, B. Husic, A. Rodriguez, C. Clementi, F. Noé, A. Laio, *Chem. Rev.* **121**(16), 9722-9758 (2021)
  - P. Gkeka *et al.*, *J. Chem. Theory Comput.* **16**(8), 4757-4775 (2020)
  - F. Noé, A. Tkatchenko, K.-R. Müller, C. Clementi, *Annu. Rev. Phys. Chem.* **71**, 361-390 (2020)
  - A.L. Ferguson, *J. Phys.: Condens. Matter* **30**, 04300 (2018)
  - M. Chen, *Eur. Phys. J. B* **94**, 211 (2021)
- **More general ML references**
  - P. Mehta, M. Bukov, C.-H. Wang, A.G.R.Day, C. Richardson, C.K. Fisher, D.J. Schwab, A high-bias, low-variance introduction to Machine Learning for physicists, *Physics Reports* **810**, 1-124 (2019)
  - I. Goodfellow, Y. Bengio, A. Courville *Deep Learning* (MIT Press, 2016)  
<http://www.deeplearningbook.org>
  - K.P. Murphy, *Probabilistic Machine Learning: An Introduction* (MIT Press, 2022)

# Molecular description of systems

# Statistical physics (1)

What is the **structure** of the protein? What are its **typical conformations**, and what are the **transition pathways** from one conformation to another?



## Statistical physics (2)

- **Microstate** of a classical system of  $N$  particles:

$$(q, p) = (q_1, \dots, q_N, p_1, \dots, p_N) \in \mathcal{E} = (a\mathbb{T})^{3N} \times \mathbb{R}^{3N}$$

**Positions**  $q$  (configuration), **momenta**  $p$  (to be thought of as  $M\dot{q}$ )

- **Hamiltonian**  $H(q, p) = V(q) + \sum_{i=1}^N \frac{p_i^2}{2m_i}$  (physics is in  $V$ )

**Macrostate:** Boltzmann–Gibbs probability measure (NVT)

$$\mu(dq dp) = Z_{\text{NVT}}^{-1} e^{-\beta H(q,p)} dq dp, \quad \beta = \frac{1}{k_{\text{B}}T}$$

- Typical evolution equations: Langevin dynamics (friction  $\gamma > 0$ )

$$\begin{cases} dq_t = M^{-1} p_t dt \\ dp_t = -\nabla V(q_t) dt - \gamma M^{-1} p_t dt + \sqrt{2\gamma\beta^{-1}} dW_t \end{cases}$$

# Reaction coordinates (RC) / collective variables (CV)

- **Reaction coordinate**  $\xi : (a\mathbb{T})^D \rightarrow \mathbb{R}^d$  with  $d \ll D$
- Ideally:  $\xi(q_t)$  captures the **slow** part of the dynamics
- **Free energy** computed on  $\Sigma(z) = \{q \in (a\mathbb{T})^D \mid \xi(q) = z\}$  (foliation)

$$F(z) = -\frac{1}{\beta} \ln \left( \int_{\Sigma(z)} e^{-\beta V(q)} \delta_{\xi(q)-z}(dq) \right)$$

- Various methods: TI, FEP, ABF, metadynamics, etc<sup>2</sup>

---

<sup>2</sup>Lelièvre/Rousset/Stoltz, *Free Energy Computations: A Mathematical Perspective* (Imperial College Press, 2010)

# ML approaches for finding CV



# Some representative approaches for finding CV (1)

- Chemical/physical **intuition** (distances, angles, RMSDs, coordination numbers, etc)
- **Short list of data-oriented approaches** (depending on the data at hand...)
  - [supervised learning] separate metastable states
  - [unsupervised/static] distinguish linear models (PCA) and nonlinear ones (e.g. based on autoencoders such as **MESA**<sup>3</sup>)
  - [unsupervised/dynamics] operator based approaches (VAC, EDMD, diffusion maps, MSM; incl. tICA and VAMPNets)

(Huge literature! I am not quoting precise references here because the list would be too long)

- Other classifications<sup>4,5</sup> possible, e.g. **slow vs. high variance CV**

---

<sup>3</sup>W. Chen and A.L. Ferguson, *J. Comput. Chem.* 2018; W. Chen, A.R. Tan, and A.L. Ferguson, *J. Chem. Phys.* 2018

<sup>4</sup>P. Gkeka et al., *J. Chem. Theory Comput.* (2020)

<sup>5</sup>A. Gliemlo et al., *Annu. Rev. Phys. Chem.* (2021)

# Some representative approaches for finding CV (2)

## Methods for Choosing Collective variables

### High-variance CVs

Principal Components  
Analysis (PCA)

Locally Linear  
Embedding (LLE)

Independent Component  
Analysis (ICA)

Laplacian and Hessian  
eigenmaps

Local tangent space  
alignment

Kernel PCA

Nonlinear PCA

Isomap

Diffusion maps

Multidimensional scaling

Semidefinite embedding/  
Maximum variance unfolding

### Available tools for CV identification

Diffusion-Map-directed MD  
(DM-d-MD)

Intrinsic Map Dynamics  
(iMapD)

Smooth and nonlinear datadriven CVs  
(SandCV)

Molecular Enhanced Sampling  
with Autoencoders (MESA)

Rewighted Autoencoded Variations  
Bayes for Enhanced Sampling (RAVE)

REinforcement Learning based on  
Adaptive samPLing (REAP)

### Slow CVs

Variational Approach to Conformational dynamics (VAC)

(extended) Dynamical Mode Decomposition ((E)DMD)

Kernel TICA

Markov State Models (MSM)

Time-lagged autoencoders (TAEs)

Time-lagged Independent Component  
Analysis (TICA)

Deep Canonical Correlation Analysis  
(DCCA)

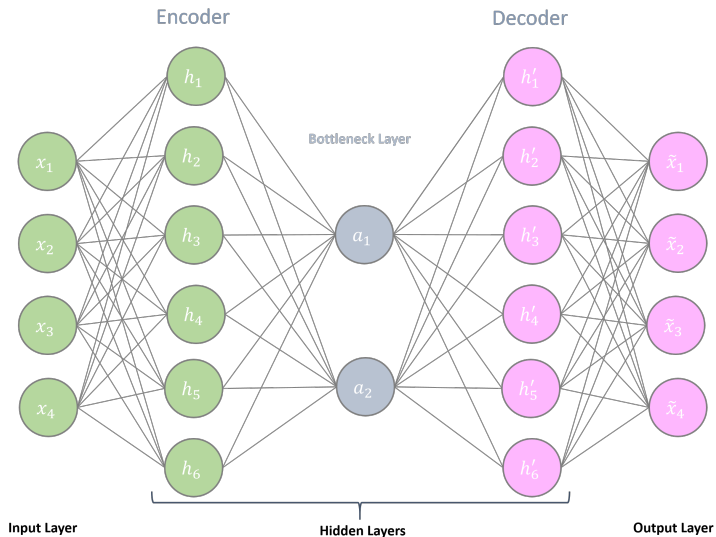
Variational Dynamics Encoders  
(VDEs)

Variational Approach for Markov Processes nets (VAMPnets)

State-free Reversible VAMPnets (SRV)

# CV construction with autoencoders

# Bottleneck autoencoders (1)



## Bottleneck autoencoders (2)

- Data space  $\mathcal{X} \subseteq \mathbb{R}^D$ , **bottleneck space**  $\mathcal{A} \subseteq \mathbb{R}^d$  with  $d < D$

$$f(x) = f_{\text{dec}}(f_{\text{enc}}(x))$$

where  $f_{\text{enc}} : \mathcal{X} \rightarrow \mathcal{A}$  and  $f_{\text{dec}} : \mathcal{A} \rightarrow \mathcal{X}$

Collective variable = encoder part

$$\xi = f_{\text{enc}}$$

- Fully connected neural network, symmetrical structure,  $2L$  layers
- Parameters  $\mathbf{p} = \{p_k\}_{k=1, \dots, K}$  (bias vectors  $b_\ell$  and weights matrices  $W_\ell$ )

$$f_{\mathbf{p}}(x) = g_{2L} [b_{2L} + W_{2L} \dots g_1 (b_1 + W_1 x)],$$

with activation functions  $g_\ell$

(examples:  $\tanh(x)$ , ReLU  $\max(0, x)$ , sigmoid  $\sigma(x) = 1/(1 + e^{-x})$ , etc)

# Training autoencoders

- **Theoretically:** minimization problem in  $\mathcal{P} \subset \mathbb{R}^K$

$$\mathbf{p}_\mu \in \operatorname{argmin}_{\mathbf{p} \in \mathcal{P}} \mathcal{L}(\mu, \mathbf{p}),$$

with **cost function**

$$\mathcal{L}(\mu, \mathbf{p}) = \mathbb{E}_\mu(\|X - f_{\mathbf{p}}(X)\|^2) = \int_{\mathcal{X}} \|x - f_{\mathbf{p}}(x)\|^2 \mu(dx)$$

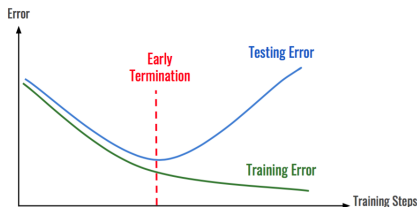
- In practice, access only to a sample: **minimization of empirical cost**

$$\mathcal{L}(\hat{\mu}, \mathbf{p}) = \frac{1}{N} \sum_{i=1}^N \|x^i - f_{\mathbf{p}}(x^i)\|^2, \quad \hat{\mu} = \frac{1}{N} \sum_{i=1}^N \delta_{x^i}$$

- **Typical choices:** canonical measure  $\mu$ , data points  $x^i$  postprocessed from positions  $q$  (alignment to reference structure, centering, reduction to backbone carbon atoms, etc)

# Some elements on training neural networks

- Many local minima...
- **Actual procedure:**
  - Separate data set into training/validation: optimize on training set
  - “Early stopping”: stop when validation loss no longer improves<sup>6</sup>



- Computation of gradient performed with backpropagation
- Choice of optimization method<sup>7</sup>, here Adam
- Add regularization to avoid overfitting (e.g.  $l^1/l^2$ , dropout, etc)

<sup>6</sup>See Section 7.8 in [Goodfellow/Bengio/Courville]

<sup>7</sup>See Chapter 8 in [Goodfellow/Bengio/Courville]

# Free energy biasing for complex CV



# Extended systems

- Computing  $\nabla\xi$  **already difficult**, higher order derivatives is worse
- **Extended system** strategy :  $V_{\text{ext}}(q, \lambda) = V(q) + \frac{\kappa}{2}(\xi(q) - \lambda)^2$
- Free energy for the (simple) **collective variable**  $\xi_{\text{ext}}(q, \lambda) = \lambda$

$$\begin{aligned} F_{\kappa}(z) &= -\frac{1}{\beta} \ln \int_{\mathcal{D}} e^{-\beta V_{\text{ext}}(q, z)} dq + C \\ &= -\frac{1}{\beta} \ln \int \left( \int_{\Sigma(\zeta)} e^{-\beta V(q)} \delta_{\xi(q) - \zeta}(dq) \right) e^{-\beta \kappa (\zeta - z)^2 / 2} d\zeta + C \\ &= -\frac{1}{\beta} \ln \int e^{-\beta F(\zeta)} \chi_{\kappa}(z - \zeta) d\zeta + \tilde{C}, \quad \chi_{\kappa}(s) = \left( \frac{\beta \kappa}{2\pi} \right)^{d/2} e^{-\beta \kappa s^2 / 2} \\ &\xrightarrow{\kappa \rightarrow +\infty} F(z) \end{aligned}$$

Calls for taking  $\kappa$  **large**

## Extended overdamped Langevin dynamics ( $\kappa$ limits $\Delta t \dots$ )

$$\begin{cases} dq_t = \left[ -\nabla V(q_t) + \kappa(\xi(q_t) - \lambda_t)\nabla \xi(q_t) \right] dt + \sqrt{2\beta^{-1}} dW_t^q \\ d\lambda_t = -\kappa[\lambda_t - \xi(q_t)] dt + \sqrt{2\beta^{-1}} dW_t^\lambda \end{cases}$$

Bias by the **free energy**: add  $F'_\kappa(\lambda) =$  steady state conditional average of  $\kappa(\lambda - \xi(q))$

## Extended ABF overdamped Langevin dynamics

$$\begin{cases} dq_t = \left[ -\nabla V(q_t) + \kappa(\xi(q_t) - \lambda_t)\nabla \xi(q_t) \right] dt + \sqrt{2\beta^{-1}} dW_t^q \\ d\lambda_t = \kappa[\xi(q_t) - \mathbb{E}(\xi(q_t) | \lambda_t)] dt + \sqrt{2\beta^{-1}} dW_t^\lambda \end{cases}$$

In practice,  $\mathbb{E}(\xi(q_t) | \lambda_t)$  is estimated by 
$$\frac{\int_0^t \delta_\varepsilon(\lambda_s - \Lambda)\xi(q_s) ds}{\max\left(\eta, \int_0^t \delta_\varepsilon(\lambda_s - \Lambda) ds\right)}$$

# Unbiased estimate of the free energy in eABF

- **Stationarity**: configurations distributed according to  $e^{-\beta(V_{\text{ext}}(q,\lambda) - F_{\kappa}(\lambda))}$

$$\rho(z, \lambda) = Z_{\kappa}^{-1} \exp \left( -\beta \left[ F(z) + \frac{\kappa}{2} (z - \lambda)^2 - F_{\kappa}(\lambda) \right] \right)$$

- **Unbiased estimator of the mean force (CZAR)<sup>8</sup>**

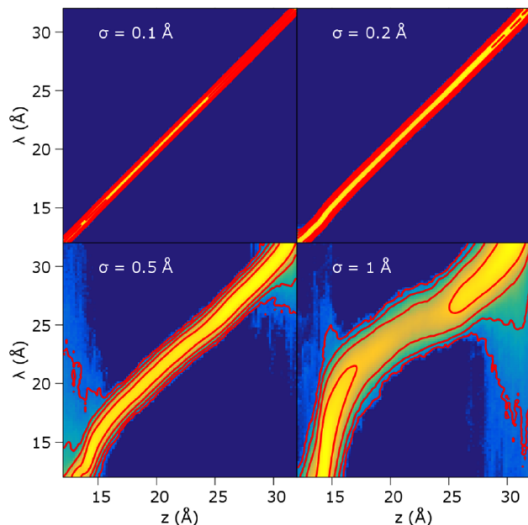
$$F'(z) = -\frac{1}{\beta} \frac{d[\ln \bar{\rho}(z)]}{dz} + \kappa(\langle \lambda \rangle_z - z)$$

with  $\bar{\rho}(z) = \int \rho(z, \lambda) d\lambda$  and  $\langle \lambda \rangle_z = \frac{1}{\bar{\rho}(z)} \int \lambda \rho(z, \lambda) d\lambda$  (**conditional dist.**)

Proof: start from  $F'(z) = -\frac{1}{\beta} \frac{\partial_z \rho(z, \lambda)}{\rho(z, \lambda)} - \kappa(z - \lambda)$ , multiply both sides of the equality by  $\rho(z, \lambda)/\bar{\rho}(z)$  and integrate with respect to  $\lambda$

<sup>8</sup>A. Lesage, T. Lelièvre, G. Stoltz and J. Hénin, *J. Phys. Chem. B* (2017)

# Joint distribution of $(\lambda, z)$ (deca-alanine)



logarithmic scale

$$\sigma^2 = \frac{1}{\beta\kappa}$$

Marginal distribution  
in  $\lambda$  nearly uniform  
(as expected)

# Iterative

free energy biasing/  
autoencoder learning

# Training on modified target measures

- Interesting systems are **metastable** (no spontaneous exploration of phase space)  
Sample according to a biased distribution  $\tilde{\mu}$  (**importance sampling**)

- Need for **reweighting** to learn the correct encoding!

$$w(x) = \frac{\mu(x)}{\tilde{\mu}(x)}$$

- **Minimization problem:** theoretical cost function

$$\mathcal{L}(\mu, \mathbf{p}) = \int_{\mathcal{X}} \|x - f_{\mathbf{p}}(x)\|^2 w(x) \tilde{\mu}(dx),$$

actual cost function

$$\mathcal{L}(\hat{\mu}_{\text{wght}}, \mathbf{p}) = \sum_{i=1}^N \hat{w}_i \|x^i - f_{\mathbf{p}}(x^i)\|^2, \quad \hat{w}_i = \frac{\mu(x^i)/\tilde{\mu}(x^i)}{\sum_{j=1}^N \mu(x^j)/\tilde{\mu}(x^j)}$$

- Only requires the knowledge of  $\mu$  and  $\tilde{\mu}$  up to a multiplicative constant.
- Stochastic gradients in training: sampling with replacement according to multinomial distribution

# Proof of concept (1)

- **Gaussian distributions**  $\mu_i = \mathcal{N}(0, \Sigma_i)$  with

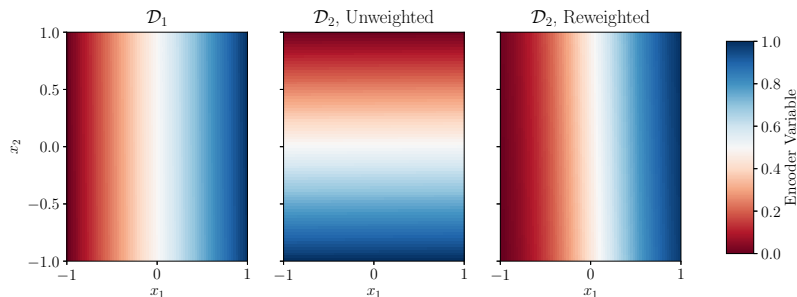
$$\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0.01 \end{pmatrix}, \quad \Sigma_2 = \begin{pmatrix} 0.01 & 0 \\ 0 & 1 \end{pmatrix}$$

Datasets  $\mathcal{D}_i$  of  $N = 10^6$  i.i.d. points

- Autoencoders with 2 layers of resp. 1 and 2 nodes, linear activation functions ( $\simeq$  PCA)
- **Training on:**
  - $\mathcal{D}_1$
  - $\mathcal{D}_2$
  - $\mathcal{D}_2$  with reweighting  $\hat{w}_i \propto \mu_1/\mu_2$

# Proof of concept (2)

## Heat maps of $f_{\text{enc}}$



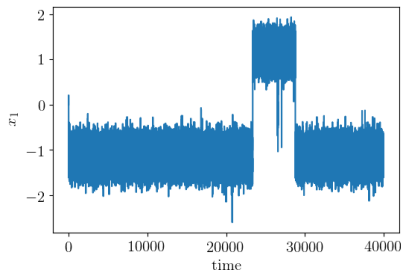
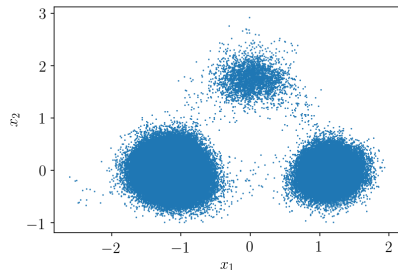
Third encoder very similar to the first one: projection on  $x_1$ .  
Second encoder projects on a direction close to  $x_2$ .



# Proof of concept with free energy biasing (1)

## Two dimensional potential (“entropic switch”)<sup>9</sup>

$$V(x_1, x_2) = 3e^{-x_1^2} \left( e^{-(x_2-1/3)^2} - e^{-(x_2-5/3)^2} \right) - 5e^{-x_2^2} \left( e^{-(x_1-1)^2} + e^{-(x_1+1)^2} \right) + 0.2x_1^4 + 0.2(x_2 - 1/3)^4$$



Trajectory from  $q^{j+1} = q^j - \nabla V(q^j)\Delta t + \sqrt{2\beta^{-1}\Delta t}G^j$  for  $\beta = 4$  and  $\Delta t = 10^{-3} \rightarrow$  **metastability** in the  $x_1$  direction

<sup>9</sup>S. Park, M.K. Sener, D. Lu, and K. Schulten (2003)

## Proof of concept with free energy biasing (2)

- **Free energy biasing:** distributions  $Z_i^{-1} \exp(-\beta [V(q) - F_i(\xi_i(q))])$

$$F_1(x_1) = -\frac{1}{\beta} \ln \left( \int_{\mathbb{R}} e^{-\beta V(x_1, x_2)} dx_2 \right), \quad F_2(x_2) = -\beta^{-1} \ln \left( \int_{\mathbb{R}} \dots dx_1 \right)$$

**Three datasets:** unbiased trajectory, trajectories biased using  $F_1$  and  $F_2$

(free energy biased trajectories are shorter but same number of data points  $N = 10^6$ )

- Autoencoders: 2-1-2 topology, activation functions  $\tanh$  (so that CV is in  $[-1, 1]$ ) then identity

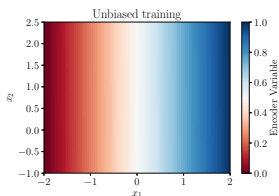
- **Five training scenarios:**

- training on long unbiased trajectory (reference CV)
- $\xi_1$ -biased trajectory, **with** or **without** reweighting
- $\xi_2$ -biased trajectory, **with** or **without** reweighting

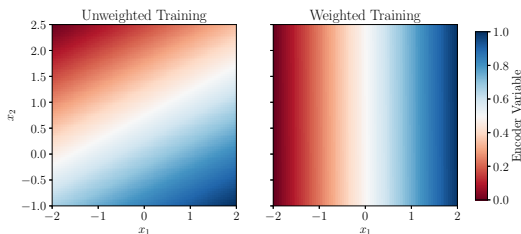
# Proof of concept with free energy biasing (3)

Normalize to compare

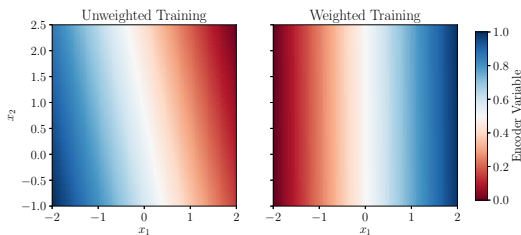
$$\xi_{\text{SAE}}^{\text{norm}}(x) = \frac{\xi_{\text{SAE}}(x) - \xi_{\text{SAE}}^{\text{min}}}{\xi_{\text{SAE}}^{\text{max}} - \xi_{\text{SAE}}^{\text{min}}}$$



Reference CV  
(distinguishes well the 3 wells)



$x_1$ -biased trajectory



$x_2$ -biased trajectory

# Full iterative algorithm (Free Energy Biasing and Iterative Learning with AutoEncoders)

**Input:** Initial condition  $q_0$ , autoencoder topology and initialization parameters  $A_{\text{init}}$ , number of samples  $N$ , simulation procedure  $S$  and **adaptive biasing procedure**  $S_{\text{AB}}$ , maximum number of iterations  $I_{\text{max}}$ , minimum convergence score  $s_{\text{min}}$

## Initialization

Sample  $\text{traj}_0 \leftarrow S(q_0, N)$

Initialize autoencoder  $\text{AE}_0 \leftarrow A_{\text{init}}$

Train  $\text{AE}_0$  on  $\text{traj}_0$  with weights  $(\hat{w}_0, \dots, \hat{w}_N) = (1, \dots, 1)$

Extract the encoder function  $\xi_0 : x \mapsto \xi_0(x)$

## Iterative update of the collective variable

Set  $i \leftarrow 0, s \leftarrow 0$

**While**  $i < I_{\text{max}}$  and  $s < s_{\text{min}}$

Set  $i \leftarrow i + 1$

Sample  $\text{traj}_i, F_i \leftarrow S_{\text{AB}}(q_0, N, \xi_{i-1})$

Compute weights  $\hat{w}_j \propto e^{-\beta F_i(\xi_{i-1}(x^j))}$

Initialize autoencoder  $\text{AE}_i \leftarrow A_{\text{init}}$

Train  $\text{AE}_i$  on  $\text{traj}_i$  with sample weights  $\hat{w}_j$

Extract the encoder function  $\xi_i : x \mapsto \xi_i(x)$

Set  $s \leftarrow \text{regscore}(\xi_{i-1}, \xi_i)$

Set  $\xi_{\text{final}} \leftarrow \xi_i$

Threshold  $s_{\text{min}}$  to be determined

in our case: extended ABF

Convergence metric to be made precise

## Production of output:

Sample  $\text{traj}_{\text{final}}, F_{\text{final}} \leftarrow S_{\text{AB}}(q_0, N_{\text{final}}, \xi_{\text{final}})$  with  $N_{\text{final}}$  large enough to ensure PMF convergence

## Discussion on the convergence criterion (1/2)

- Check convergence of CV?

Quantify  $\xi_i \approx \Phi(\xi_{i-1})$  for some **monotonic function**  $\Phi$

- Approach: approximate  $\Phi$  by a **linear model** (Nonlinear regression may be needed)

- **Regression score** between  $\xi$  and  $\xi'$

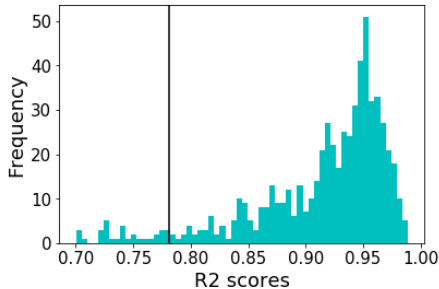
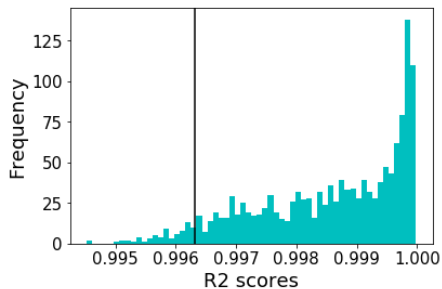
- Two sets of values of CV ( $\xi(q^1), \dots, \xi(q^N)$ ) and ( $\xi'(q^1), \dots, \xi'(q^N)$ )
- Match them with a linear model  $M(z) = Wz + b$

- Coefficient of determination  $R^2 = 1 - \frac{\sum_{i=1}^N \|\xi'(q^i) - M(\xi(q^i))\|^2}{\sum_{i=1}^N \|\xi'(q^i) - \bar{\xi}'\|^2}$

- Maximization of  $R^2$  w.r.t.  $W, b$  provides  $\text{regscore}(\xi', \xi)$

- **Value of**  $s_{\min}$  computed using some **bootstrap** procedure

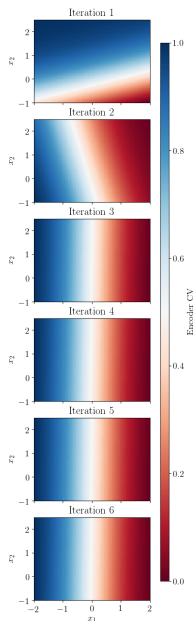
## Discussion on the convergence criterion (2/2)



Histogram of the  $R^2$  scores obtained using subsets of  $N = 10^5$  points out of  $10^6$  points (vertical black line = 5% percentile).

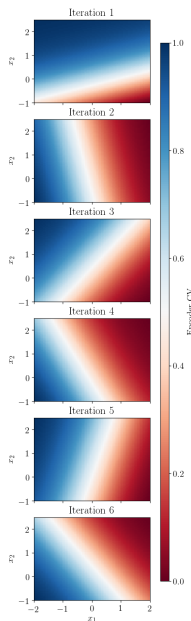
(Left: Alanine dipeptide. Right: Chignolin)

# The iterative algorithm on the toy 2D example



**Left:** with reweighting  
Convergence to CV  $\simeq x_1$

**Right:** without reweighting  
No convergence  
(cycles between two CVs)



# Applications to systems of interest



# Alanine dipeptide

- **Molecular dynamics:**

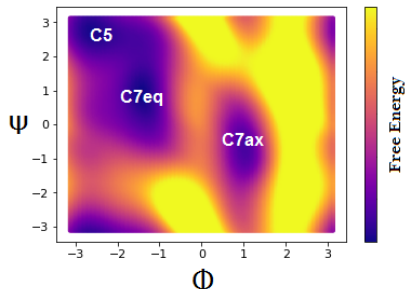
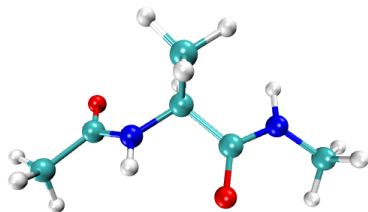
openmm with openmm-plumed to link it with plumed colvar module for eABF and computation of free energies<sup>10</sup>  
timestep 1 fs, friction  $\gamma = 1 \text{ ps}^{-1}$  in Langevin dynamics

- **Machine learning:**

keras for autoencoder training

input = carbon backbone (realignment to reference structure and centering)

neural network: topology 24-40-2-40-24, tanh activation functions

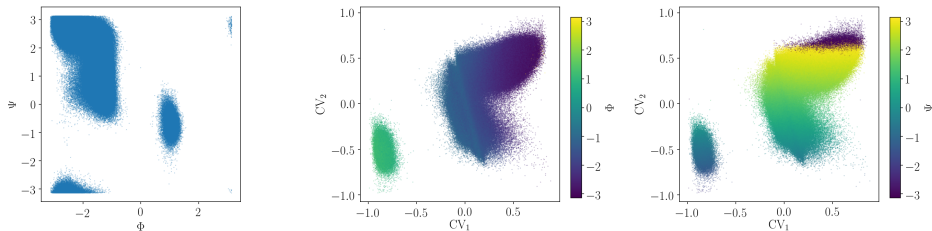


<sup>10</sup>See also Chen/Liu/Feng/Fu/Cai/Shao/Chipot, *J. Chem. Inf. Model.* (2022)

# Ground truth computation

Long trajectory ( $1.5 \mu\text{s}$ ),  $N = 10^6$  (frames saved every 1.5 ps)

CV close to dihedral angles  $\Phi, \Psi$

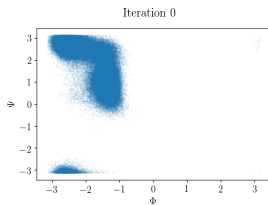


Quantify  $s_{\min} = 0.99$  for  $N = 10^5$  using a bootstrapping procedure

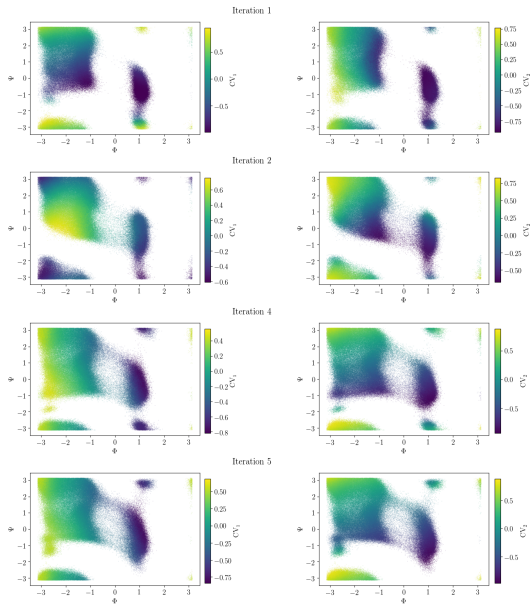
**For the iterative algorithm:** 10 ns per iteration

(compromise between times not too short to allow for convergence of the free energy, and not too large in order to alleviate the computation cost)

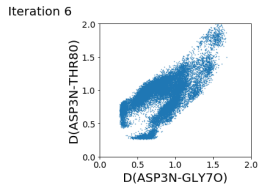
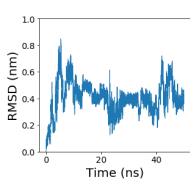
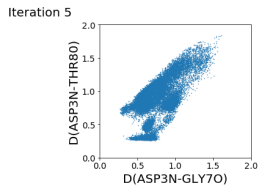
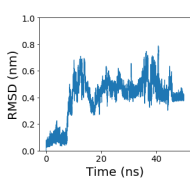
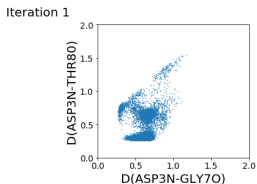
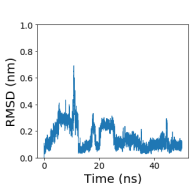
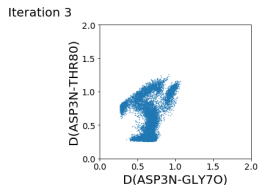
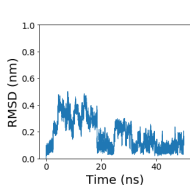
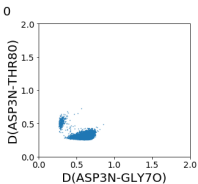
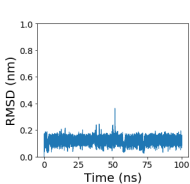
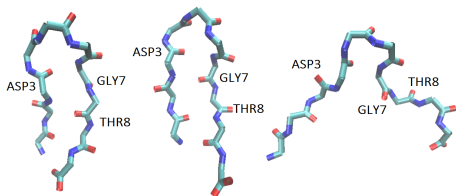
# Results for the iterative algorithm



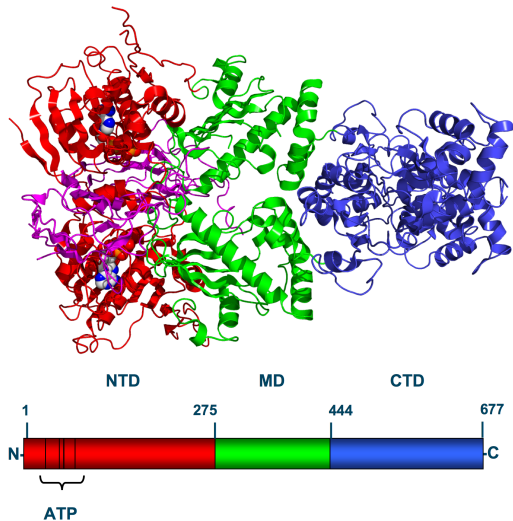
iter.	regscore	$(\Phi, \Psi)$
0	—	0.922
1	0.872	0.892
2	0.868	0.853
3	0.922	0.973
4	0.999	0.972
5	0.999	0.970
6	0.999	0.971
7	0.999	0.967
8	0.998	0.966
9	0.999	0.968



# Chignolin (Folded/misfolded/unfolded states)



# HSP90 (work in progress...)

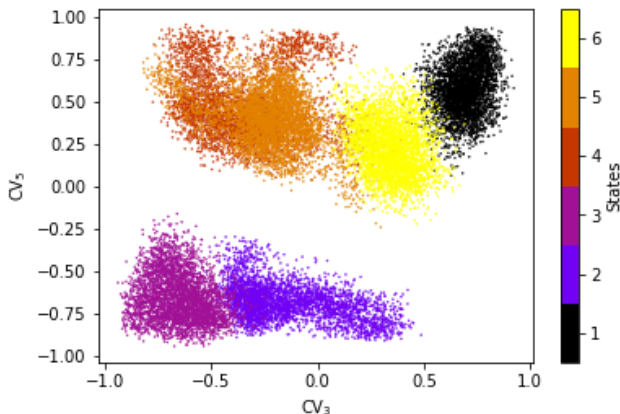


Chaperone protein assisting other proteins to fold properly and stabilizing them against stress, including proteins required for **tumor growth**

→ look for **inhibitors** (e.g. targeting binding region of ATP; focus only on the N-terminal domain )

(picture from [https://en.wikipedia.org/wiki/File:Hsp90\\_schematic\\_2cg9.png](https://en.wikipedia.org/wiki/File:Hsp90_schematic_2cg9.png))

# HSP90 (work in progress...)



6 conformational states, data from  $10 \times 20$  ns trajectories, input features = 207 C carbons, AE topology 621-100-5-100-621

**Issue: dimension of bottleneck?**

# Some perspectives

# Some perspectives

- **Incorporating knowledge/information on the transition states?**
  - compute mean square error with respect to another distribution?
  - add terms to the loss function? (e.g. related to MEP<sup>11</sup>)
- **Incorporating dynamical information?**
  - time-lagged autoencoders and their variations<sup>12</sup>
  - making use of the generator of the dynamics<sup>13</sup>
- **Better understanding autoencoders**
  - choice of topology: mathematical analysis?
  - simple dimensionality reduction methods in the bottleneck (to allow for free energy computations)

---

<sup>11</sup>Ramil/Boudier/Gorayeva/Marinica/Maillet, *J. Chem. Theory Comput.* (2022)

<sup>12</sup>Chen/Sidky/Ferguson, *J. Chem. Phys.* (2019)

<sup>13</sup>Zhang/Li/Schütte, *J. Comput. Phys.* (2022)