Coarse-graining and efficiently sampling with autoencoders

Gabriel STOLTZ

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

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

BRIN "Rare events" workshop, March 1st, 2023

Outline

- **A (short/biased) review of machine learning approaches for CV**

- **Free-energy biasing and iterative learning with autoencoders**[1]
  - 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)

[1] Z. Belkacemi, P. Gkeka, T. Lelièvre, G. Stoltz, *J. Chem. Theory Comput.* **18** (2022)

(A biased perspective on some) References

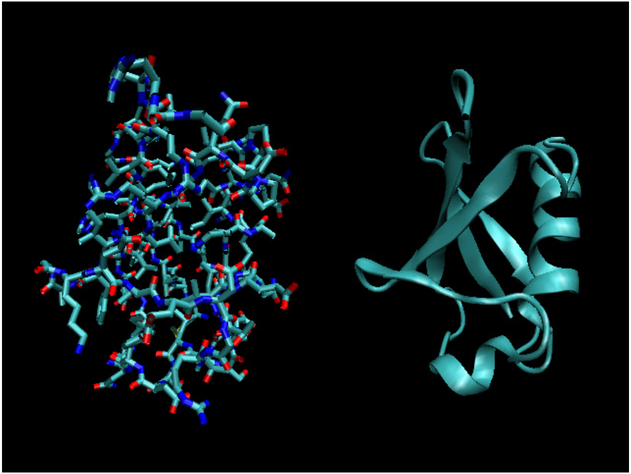
- **ML reviews in MD** (biased towards dimensionality reduction, not force fields)
  - A. Gliemlo, B. Husic, A. Rodriguez, C. Clementi, F. Noé, A. Laio, *Chem. Rev.* **121**(16), 9722-9758 (2021)
  - P. Gkeka *et al.*, *J. Chem. Theory Comput.* **16**(8), 4757-4775 (2020)
  - F. Noé, A. Tkatchenko, K.-R. 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)

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

- 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 coodinates (RC) / collective variables (CV)

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

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

- Various methods: TI, FEP, ABF, metadynamics, etc[2]

[2] 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[3])
  - [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[4,5] possible, e.g. **slow vs. high variance CV**

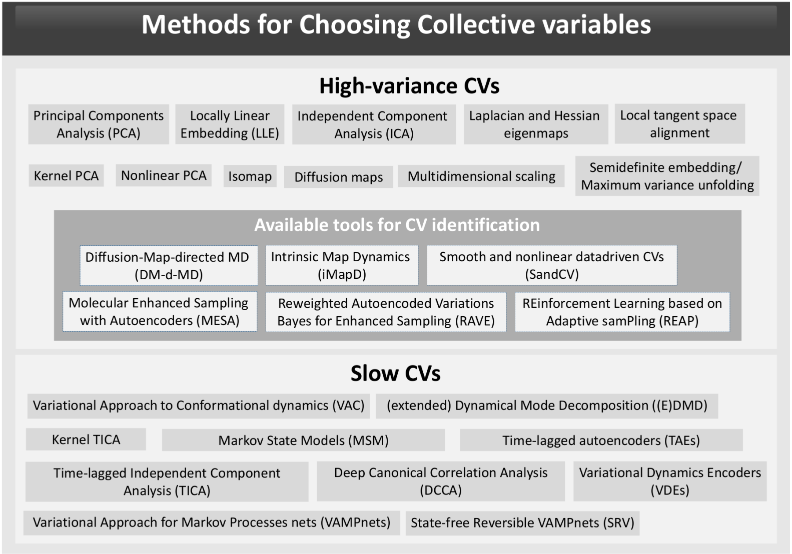
---

[3] W. Chen and A.L. Ferguson, *J. Comput. Chem.* 2018; W. Chen, A.R. Tan, and A.L. Ferguson, *J. Chem. Phys.* 2018

[4] P. Gkeka et al., *J. Chem. Theory Comput.* (2020)

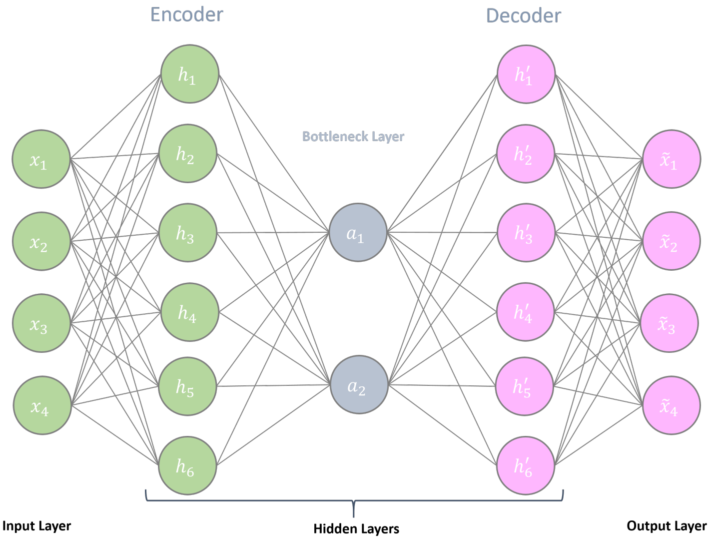
[5] A. Gliemlo *et al.*, *Annu. Rev. Phys. Chem.* (2021)

Some representative approaches for finding CV (2)



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 $d < D$

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

where $f_{\mathsf{enc}} : \mathcal{X} \to \mathcal{A}$ and $f_{\mathsf{dec}} : \mathcal{A} \to \mathcal{X}$

Collective variable = encoder part

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

- Fully connected neural network, symmetrical structure, $2L$ layers

- Parameters $\mathbf{p} = \{p_k\}_{k=1,\dots,K}$ (bias vectors b_ℓ and weights matrices W_ℓ)

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

with activation functions g_ℓ

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

Training autoencoders

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

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

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**

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

• **Typical choices:** canonical measure μ, data points x^i postprocessed from positions q (alignement to reference structure, centering, reduction to backbone carbon atoms, etc)

A variance interpretation of autoencoders

- Total variance $\boxed{\mathrm{Var}(X) = \mathrm{Var}\left[\mathbb{E}(X|f_{\mathrm{enc}}(X))\right] + \mathbb{E}\left[\mathrm{Var}(X|f_{\mathrm{enc}}(X))\right]}$

- Training w.r.t. decoder part performed analytically in principle as

$$\min_{f_{\mathrm{enc}}} \left[\min_{f_{\mathrm{dec}}} \int_{\mathcal{X}} \left| x - f_{\mathrm{dec}}(f_{\mathrm{enc}}(x)) \right|^2 \mu(dx) \right] = \min_{f_{\mathrm{enc}}} \ell(f_{\mathrm{enc}})$$

with $$\ell(f_{\mathrm{enc}}) = \mathbb{E}\left[\Big(X - \mathbb{E}(X|f_{\mathrm{enc}}(X)) \Big)^2 \right] = \mathbb{E}\left[\mathrm{Var}(X|f_{\mathrm{enc}}(X))\right] = \mathrm{Var}(X) - \mathrm{Var}\left[\mathbb{E}(X|f_{\mathrm{enc}}(X))\right]$$

- Minimizing $\ell(f_{\mathrm{enc}})$ equivalent to maximizing $\mathrm{Var}\left[\mathbb{E}(X|f_{\mathrm{enc}}(X))\right]$

$=$ minimizing intraclass dispersion vs. maximizing interclass dispersion

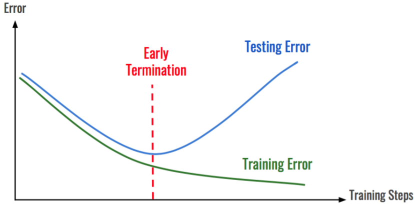
$=$ small spread of data points for f_{enc} given around the mean vs. the mean values associated with f_{enc} given should be as spread out as possible

- Principal curve interpretation, as for string method[6]

[6] Venturoli/Vanden–Eijnden (2009), Gerber/Whitaker (2013), Gerber (2021)

Some elements on training neural networks

- Many local minima...
- **Actual procedure:**
  - "Early stopping": stop when validation loss no longer improves[7]



  - Choice of optimization method[8], here Adam
  - No added regularization here (e.g. ℓ^1/ℓ^2, dropout, etc)

[7] See Section 7.8 in [Goodfellow/Bengio/Courville]
[8] 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 : $V_{\mathrm{ext}}(q,\lambda) = V(q) + \dfrac{\kappa}{2}\big(\xi(q)-\lambda\big)^2$

- Free energy for the (simple) collective variable $\xi_{\mathrm{ext}}(q,\lambda) = \lambda$

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

Calls for taking κ large

Extended ABF

Extended overdamped Langevin dynamics (κ limits Δt...)

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

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

Extended ABF overdamped Langevin dynamics

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

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

Unbiased estimate of the free energy in eABF

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

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

- **Unbiased estimator of the mean force (CZAR)**[9]

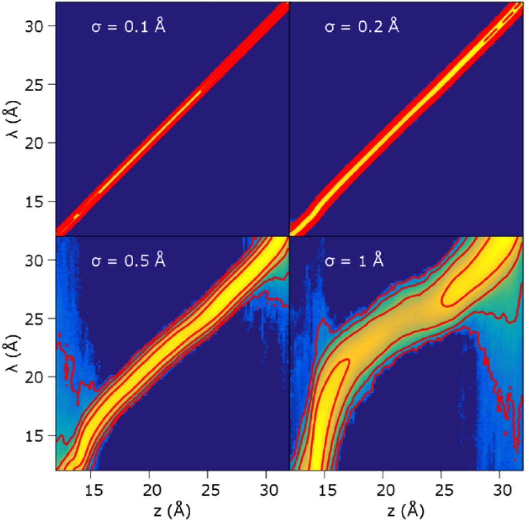
$$F'(z) = -\frac{1}{\beta}\frac{d[\ln\overline{\rho}(z)]}{dz} + \kappa(\langle\lambda\rangle_z - z)$$

with $\overline{\rho}(z) = \int \rho(z,\lambda)\,d\lambda$ and $\langle\lambda\rangle_z = \frac{1}{\overline{\rho}(z)}\int \lambda\rho(z,\lambda)\,d\lambda$ (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 $\rho(z,\lambda)/\overline{\rho}(z)$ and integrate with respect to λ

[9] A. Lesage, T. Lelièvre, G. Stoltz and J. Hénin, *J. Phys. Chem. B* (2017)

Joint distribution of (λ, z) (deca-alanine)



logarithmic scale

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

Marginal distribution in λ 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 $\widetilde{\mu}$ (importance sampling)

- Need for reweighting[10] $w(x) = \mu(x)/\widetilde{\mu}(x)$

- **Minimization problem:** theoretical cost function

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

actual cost function

$$\mathcal{L}(\widehat{\mu}_{\mathsf{wght}}, \mathbf{p}) = \sum_{i=1}^{N} \widehat{w}_i \|x^i - f_{\mathbf{p}}(x^i)\|^2, \qquad \widehat{w}_i = \frac{\mu(x^i)/\widetilde{\mu}(x^i)}{\sum_{j=1}^{N} \mu(x^j)/\widetilde{\mu}(x^j)}$$

- Only requires the knowledge of μ and $\widetilde{\mu}$ up to a multiplicative constant.

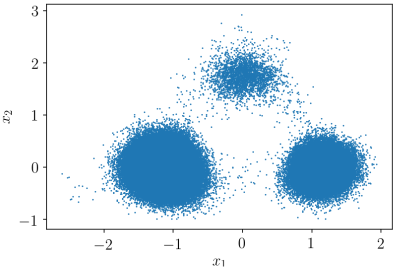
- Minibatching: multinomial distribution for sampling with replacement

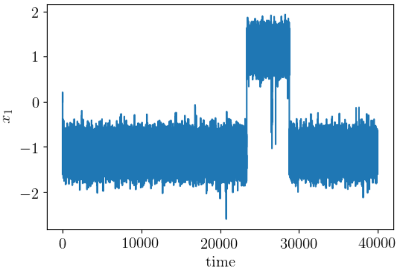
[10] As done in RAVE for instance, see Ribeiro/Bravo/Wang/Tiwary (2018), Wang/Ribeiro/Tiwary (2019)

Proof of concept with free energy biasing (1)

Two dimensional potential ("entropic switch")[11]

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





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 $\Delta t = 10^{-3}$ $\longrightarrow$ metastability in the x_1 direction

[11] S. Park, M.K. Sener, D. Lu, and K. Schulten (2003)

Proof of concept with free energy biasing (2)

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

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

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

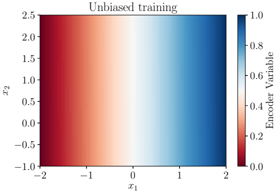
• **Five training scenarios:**

- training on long unbiased trajectory (reference CV)
- ξ_1-biased trajectory, with or without reweighting
- ξ_2-biased trajectory, with or without reweighting

Proof of concept with free energy biasing (3)

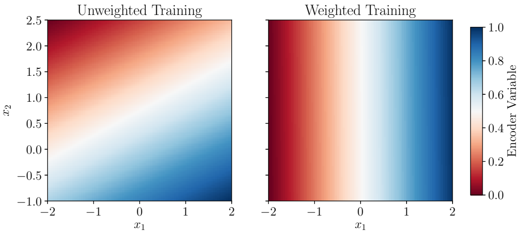
Normalize to compare

$$\xi_{\mathsf{AE}}^{\mathsf{norm}}(x) = \frac{\xi_{\mathsf{AE}}(x) - \xi_{\mathsf{AE}}^{\min}}{\xi_{\mathsf{AE}}^{\max} - \xi_{\mathsf{AE}}^{\min}}$$

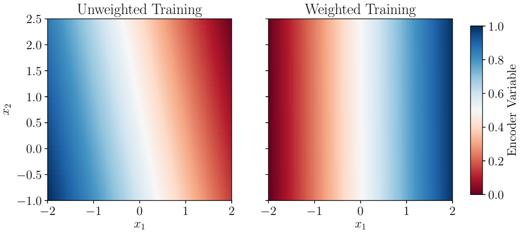


Reference CV

(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_{init}, number of samples N, simulation procedure S and adaptive biasing procedure S_{AB}, maximum number of iterations I_{max}, minimum convergence score s_{min}

Initialization
Sample $\text{traj}_0 \leftarrow S(q_0, N)$
Initialize autoencoder $\text{AE}_0 \leftarrow A_{\text{init}}$
Train AE_0 on traj_0 with weights $(\widehat{w}_0, \ldots, \widehat{w}_N) = (1, \ldots 1)$
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 $i < I_{\max}$ and $s < s_{\min}$ — Threshold $s_{\min}$ to be determined
  Set $i \leftarrow i + 1$
  Sample traj_i, $F_i \leftarrow S_{\text{AB}}(q_0, N, \xi_{i-1})$ — in our case: extended ABF
  Compute weights $\widehat{w}_j \propto \mathrm{e}^{-\beta F_i(\xi_{i-1}(x^j))}$
  Initialize autoencoder $\text{AE}_i \leftarrow A_{\text{init}}$
  Train AE_i on traj_i with sample weights $\widehat{w}_j$
  Extract the encoder function $\xi_i : x \mapsto \xi_i(x)$
  Set $s \leftarrow \text{regscore}(\xi_{i-1}, \xi_i)$ — Convergence metric to be made precise
Set $\xi_{\text{final}} \leftarrow \xi_i$

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_{final} large enough to ensure PMF convergence

Discussion on the convergence criterion (1/2)

- Check convergence of CV?

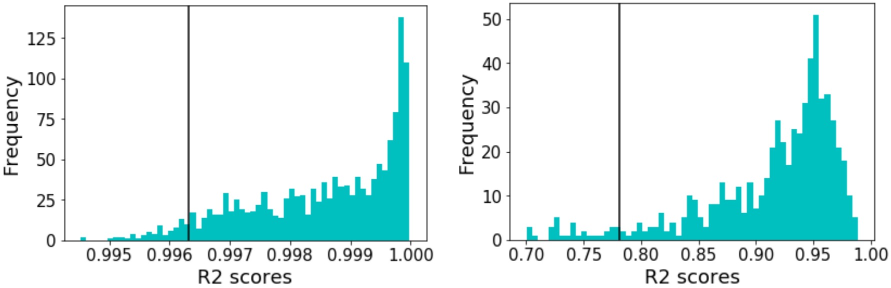
Quantify $\xi_i \approx \Phi(\xi_{i-1})$ for some monotonic function Φ

- Approach: approximate Φ by a linear model (Nonlinear regression may be needed)

- **Regression score** between ξ and ξ'
  - Two sets of values of CV $(\xi(q^1), \ldots, \xi(q^N))$ and $(\xi'(q^1), \ldots, \xi'(q^N))$
  - Match them with a linear model $M(z) = Wz + b$
  - Coefficient of determination $R^2 = 1 - \dfrac{\displaystyle\sum_{i=1}^N \left\| \xi'(q^i) - M(\xi(q^i)) \right\|^2}{\displaystyle\sum_{i=1}^N \left\| \xi'(q^i) - \bar{\xi'} \right\|^2}$
  - Maximization of R^2 w.r.t. W, b provides regscore(ξ', ξ)

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

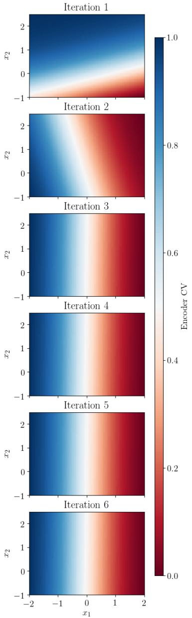
Discussion on the convergence criterion (2/2)



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

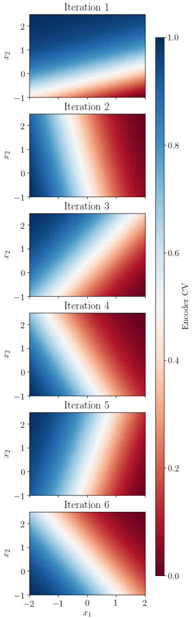
(Left: Alanine dipeptide. Right: Chignolin)

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)



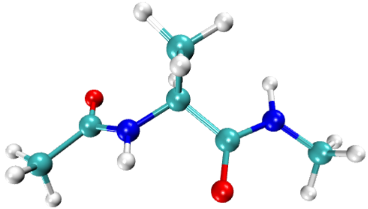
Applications to systems of interest

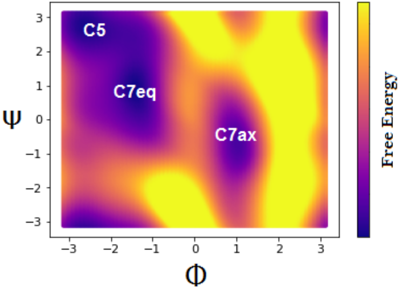
- **Molecular dynamics:**

`openmm` with `openmm-plumed` to link it with `plumed`
`colvar` module for eABF and computation of free energies[12]
timestep 1 fs, friction $\gamma = 1$ ps^{-1} in Langevin dynamics

- **Machine learning:**

`keras` for autoencoder training
input = carbon backbone (realignement to reference structure and centering)
neural network: topology 24-40-2-40-24, tanh activation functions



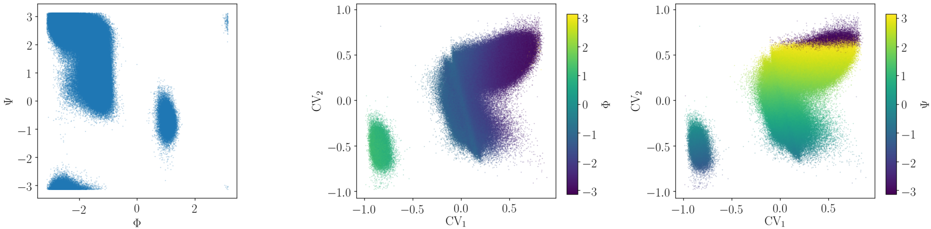


[12] See also Chen/Liu/Feng/Fu/Cai/Shao/Chipot, *J. Chem. Inf. Model.* (2022)

Ground truth computation

Long trajectory (1.5 μs), $N = 10^6$ (frames saved every 1.5 ps)

CV close to dihedral angles Φ, Ψ

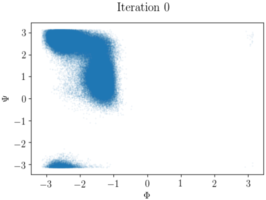


Quantify $s_{\min} = 0.99$ for $N = 10^5$ using a bootstraping procedure

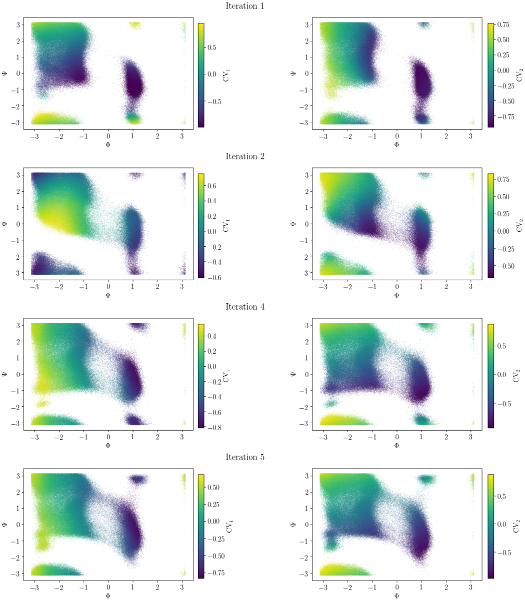
For the iterative algorithm: 10 ns per iteration

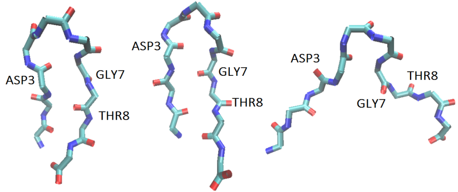
(compromise between times not too short to allow for convergence of the free energy, and not too large in order to alleviate the computation cost)

Results for the iterative algorithm

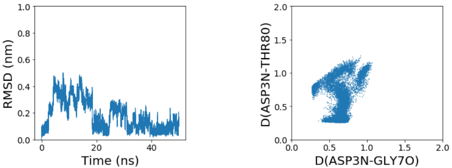


iter.	regscore	(Φ, Ψ)
0	–	0.922
1	0.872	0.892
2	0.868	0.853
3	0.922	0.973
4	0.999	0.972
5	0.999	0.970
6	0.999	0.971
7	0.999	0.967
8	0.998	0.966
9	0.999	0.968

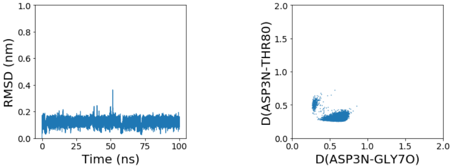




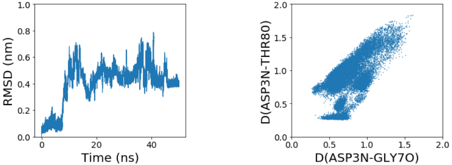
Iteration 3



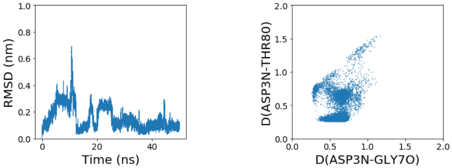
Iteration 0



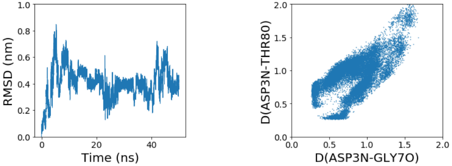
Iteration 5



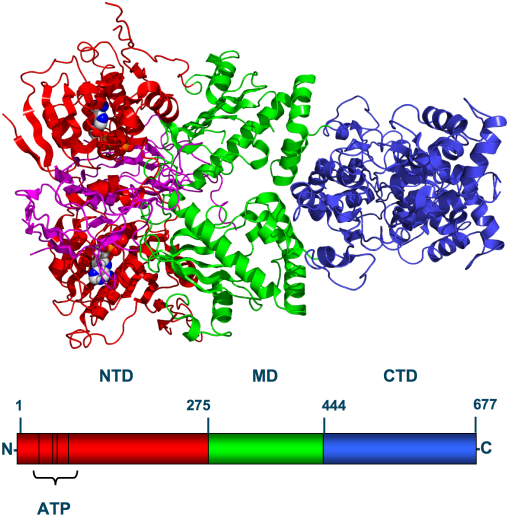
Iteration 1



Iteration 6



HSP90 (work in progress...)

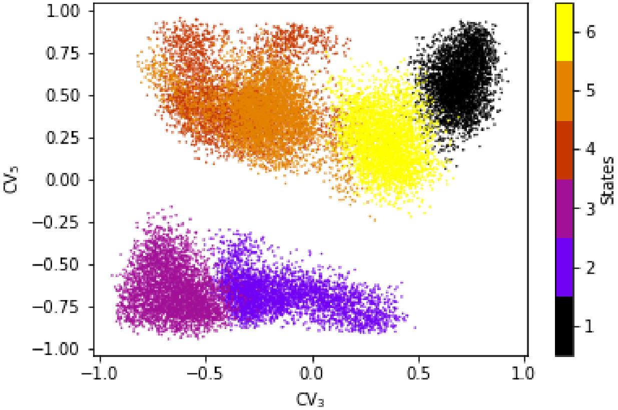


Chaperone protein assisting other proteins to fold properly and stabilizing them against stress, including proteins required for tumor growth

$\longrightarrow$ 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)

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[13])

- **Incorporating dynamical information?**
  - time-lagged autoencoders and their variations[14]
  - making use of the generator of the dynamics[15]

- **Better understanding autoencoders**
  - choice of topology: mathematical analysis?
  - simple dimensionality reduction methods in the bottleneck (to allow for free energy computations)

---

[13] Ramil/Boudier/Gorayeva/Marinica/Maillet, *J. Chem. Theory Comput.* (2022)
[14] Chen/Sidky/Ferguson, *J. Chem. Phys.* (2019)
[15] Zhang/Li/Schütte, *J. Comput. Phys.* (2022)

Interest in interfaces between ML and MD?

Program March-May 2024 at University of Chicago