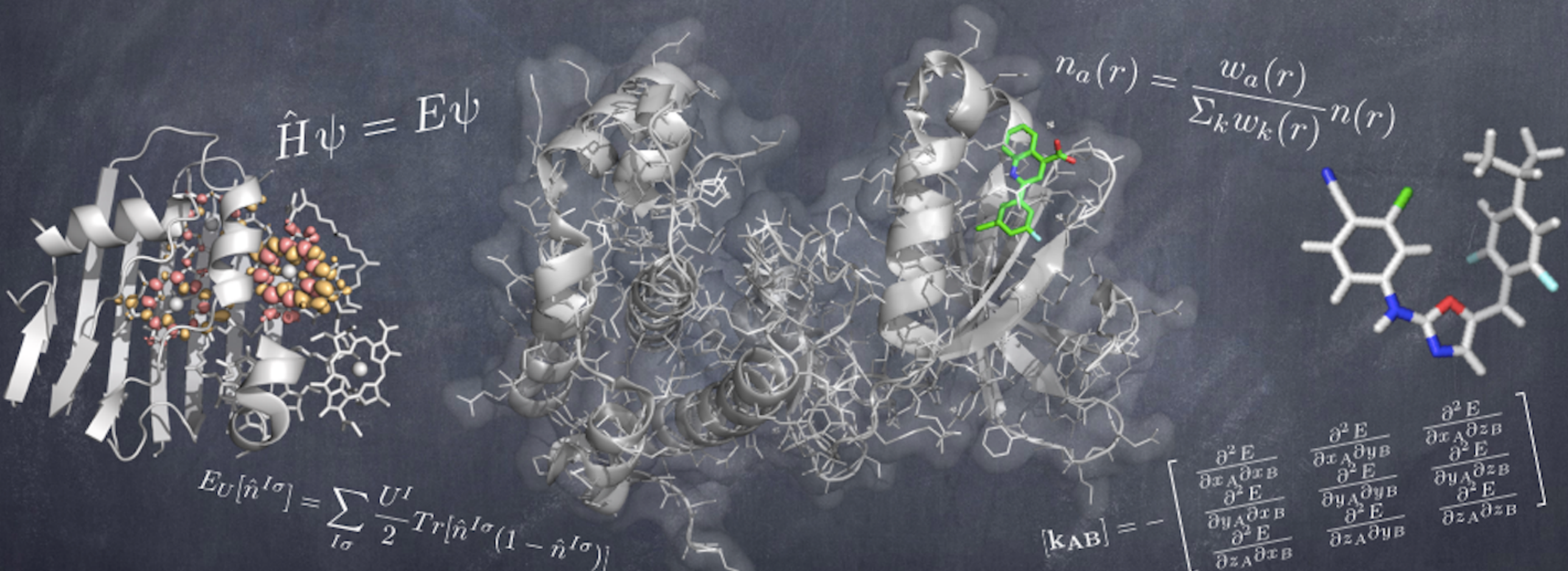



# Molecular Modelling in Medicinal Chemistry

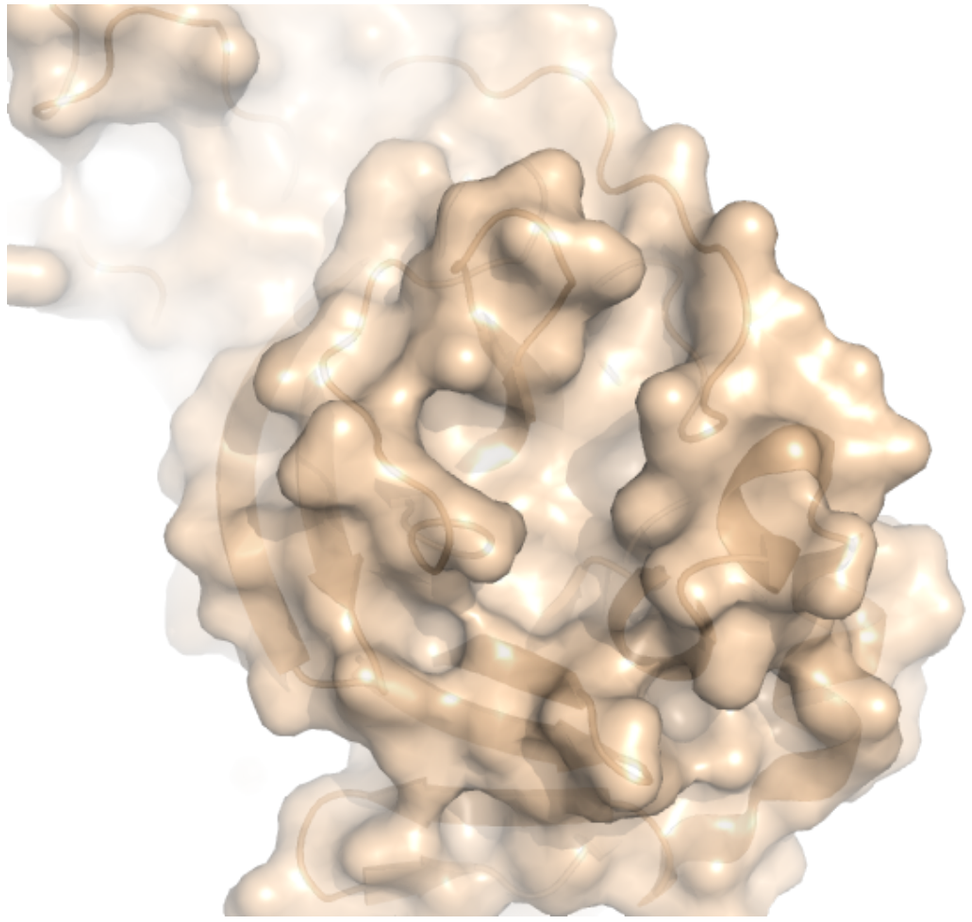
**Daniel Cole**
School of Natural and Environmental Sciences

Newcastle University

# Problem: Organic Synthesis is Costly

## Computer-Aided Drug Design

- **Medicinal Chemistry**
  – Aim to design a molecule that binds to a target (usually a protein) with therapeutic benefit



- **Medicinal Chemistry is Expensive**
  – Estimated R&D costs of $2bn

- **Multiple properties must be optimised at once**
  – Strength of binding (the right shape, suitable intermolecular interactions, …)
  – Pharmacokinetics (including absorption, distribution, metabolism and excretion properties)
  – Suitable for computational design

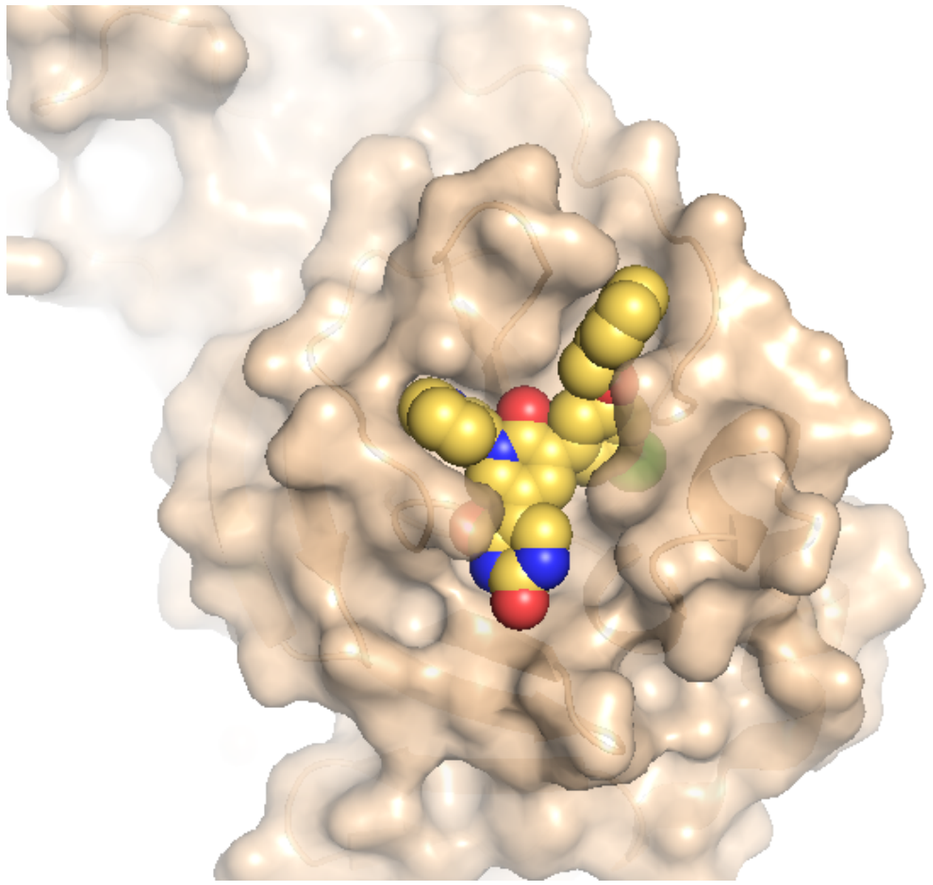
# Problem: Organic Synthesis is Costly

## Computer-Aided Drug Design

- **Medicinal Chemistry**
  – Aim to design a molecule that binds to a target (usually a protein) with therapeutic benefit

- **Medicinal Chemistry is Expensive**
  – Estimated R&D costs of $2bn

- **Multiple properties must be optimised at once**
  – Strength of binding (the right shape, suitable intermolecular interactions, …)
  – Pharmacokinetics (including absorption, distribution, metabolism and excretion properties)
  – Suitable for computational design



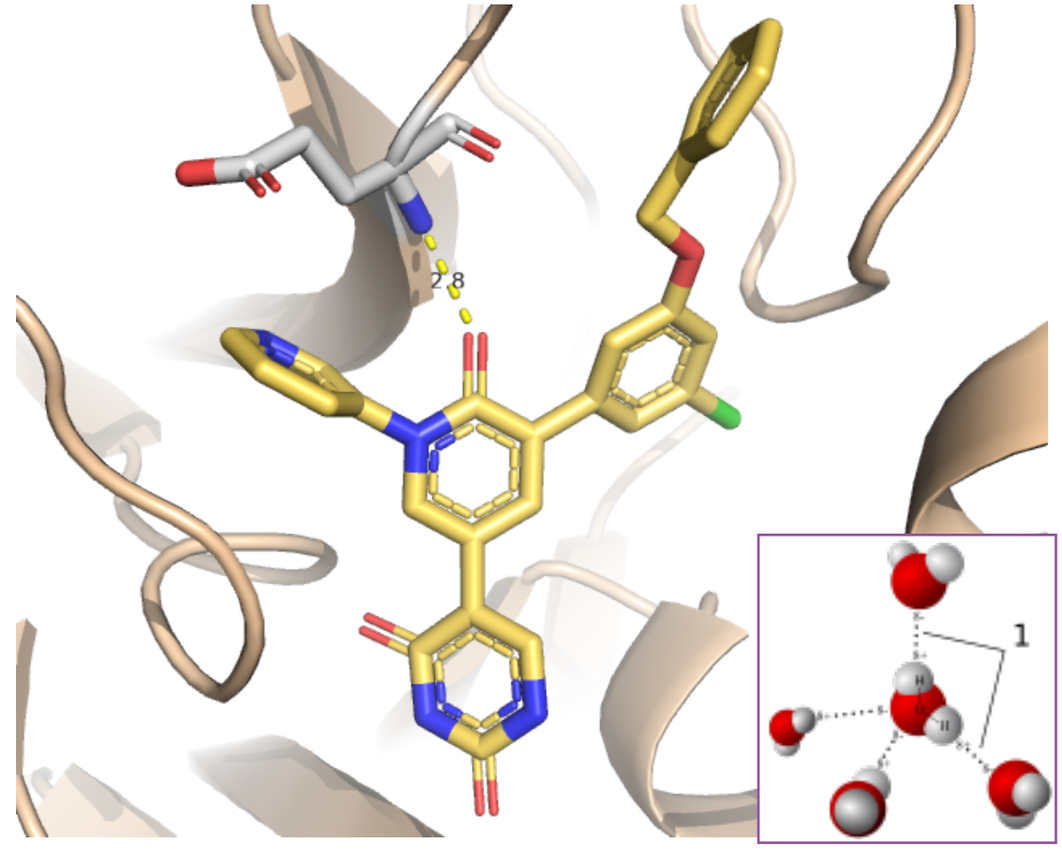
# Problem: Organic Synthesis is Costly

## Computer-Aided Drug Design

- **Medicinal Chemistry**
  – Aim to design a molecule that binds to a target (usually a protein) with therapeutic benefit

- **Medicinal Chemistry is Expensive**
  – Estimated R&D costs of $2bn

- **Multiple properties must be optimised at once**
  – Strength of binding (the right shape, suitable intermolecular interactions, …)
  – Pharmacokinetics (including absorption, distribution, metabolism and excretion properties)
  – Suitable for computational design

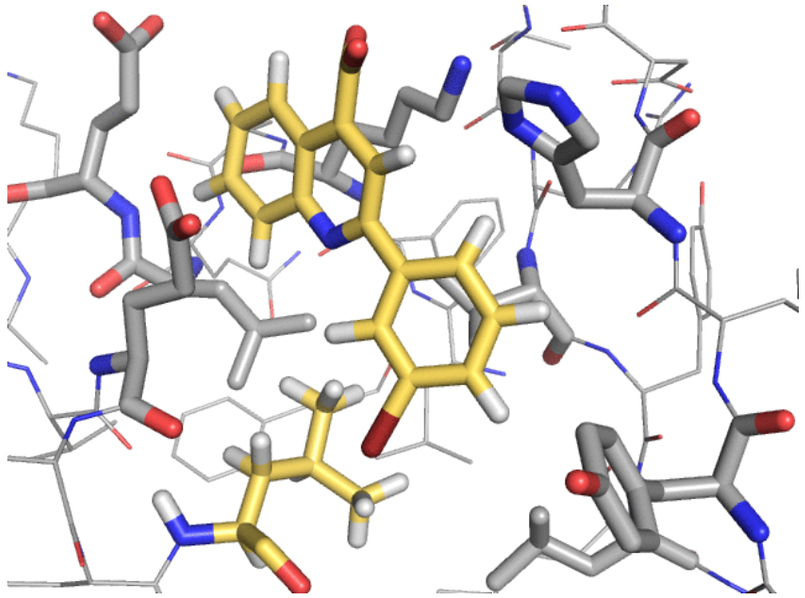



*Stabilised by e.g. hydrogen bonding interactions with the target*

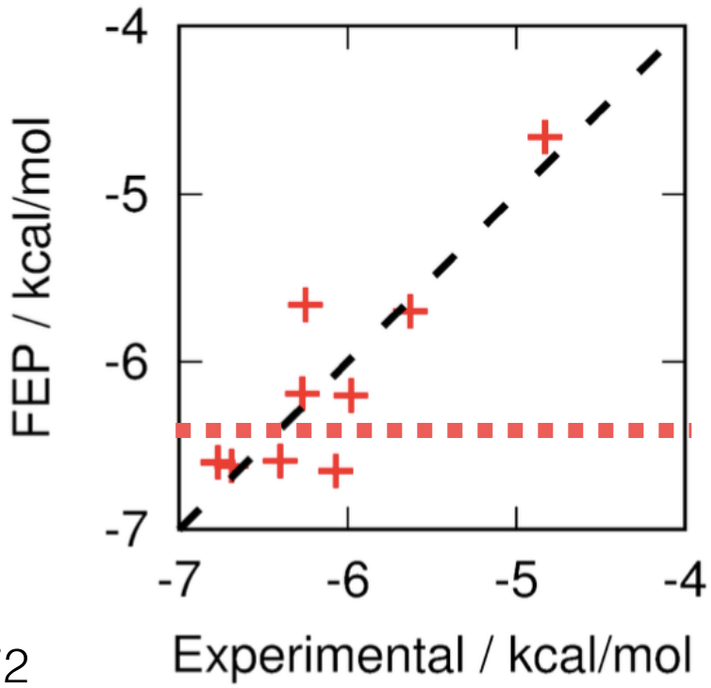
Need a computational model that can predict binding from structure

# Molecular Interactions & Dynamics

Free energy calculations based on molecular modelling fit this need.



Cole et al., *Chem. Commun.* **2017**, *53*, 9372



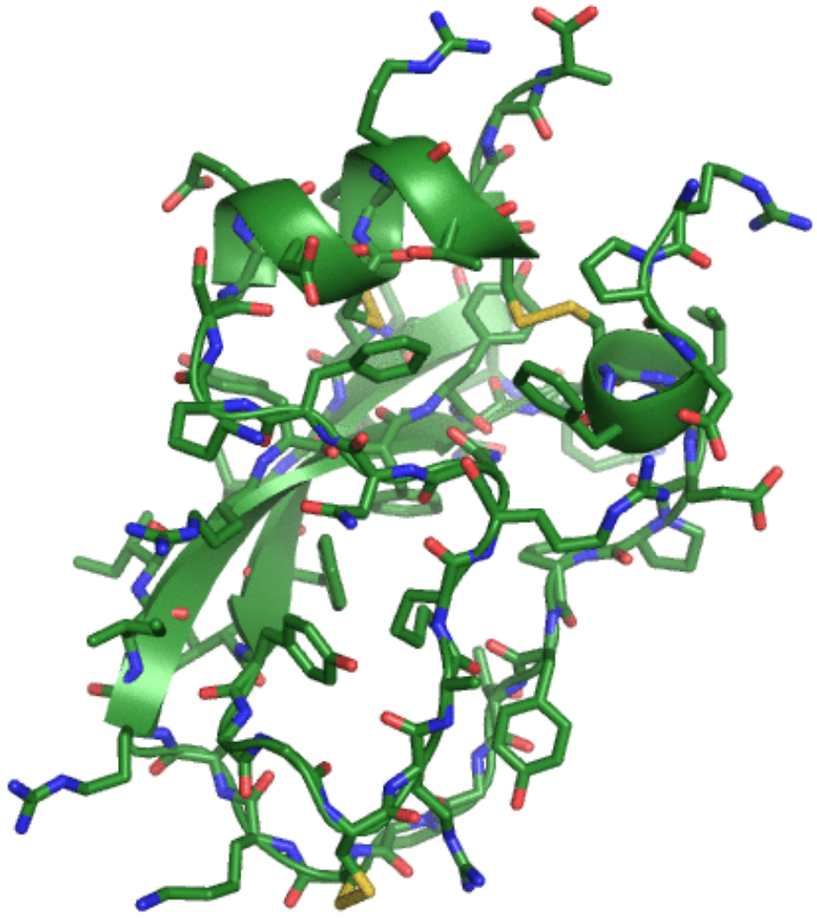

**RMSE = 0.32 kcal/mol**

Allow us to prioritise molecules for synthesis.

Methods are based around **molecular mechanics** and **force fields**…

# Molecular Mechanics

Molecular mechanics (MM) can complement experiment in many ways. At the simplest level, we can use molecular dynamics to 'animate' the system.



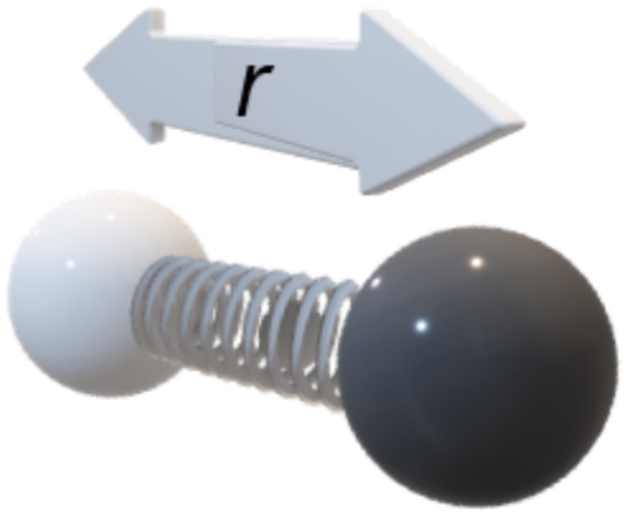
**How does it work?** If we know the structure at t=0 and the forces on the atoms, we can integrate Newton's 2nd law to find the positions at all later times:

$$\mathbf{F}_i = m_i \mathbf{a}_i = m \frac{d^2 \mathbf{r}_i}{dt^2}$$

Time steps of 1fs ($10^{-15}$ s) are required to model bond vibrations.

Simulations tend to be limited only by finite sampling (μs to ms) and accuracy of the forces.

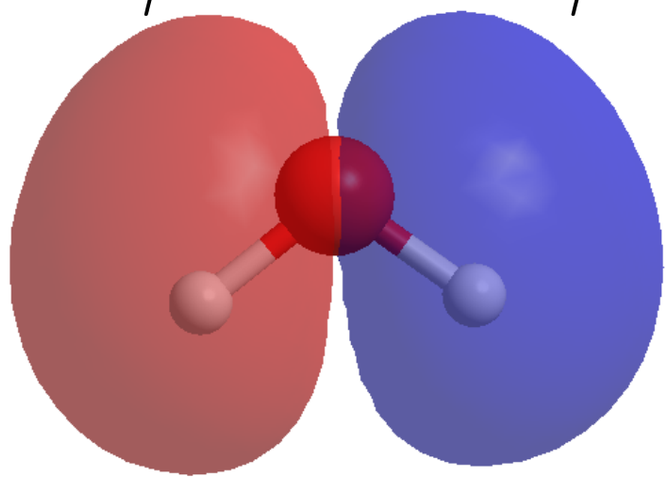
# Atomistic Modelling




**Classical mechanics**

fast to run, large system sizes, not very accurate

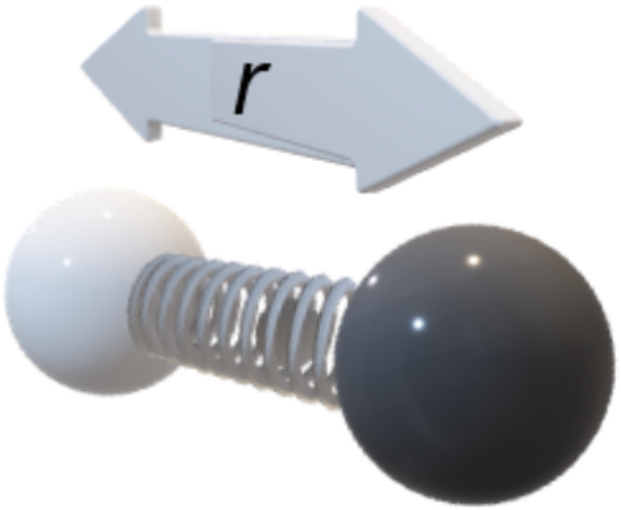
$$H\psi = E\psi$$



**Quantum mechanics**

very accurate, small system sizes, very expensive to run

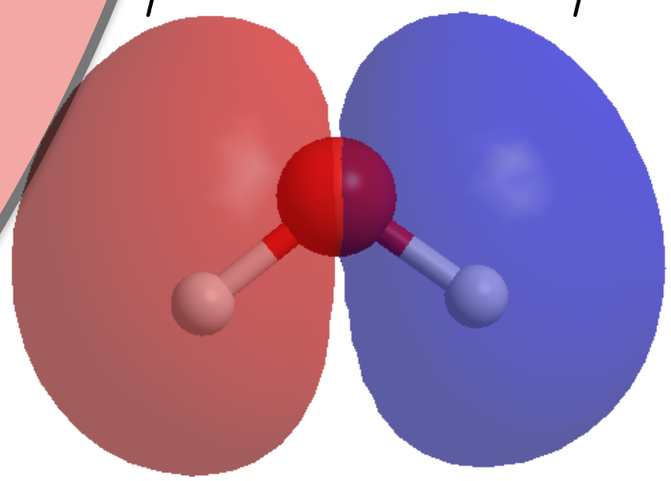
# Atomistic Modelling




**Classical mechanics**
fast to run, large system sizes, not very accurate

Not much available in this space!

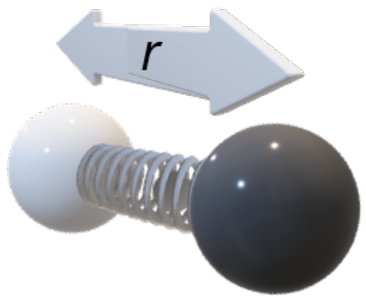
$$\hat{H}\psi = E\psi$$

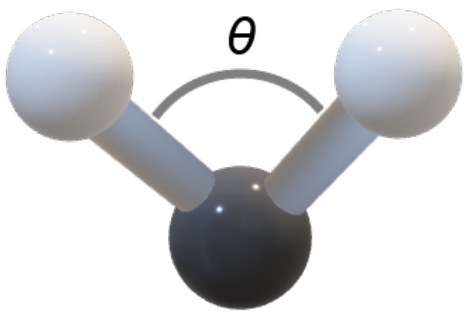


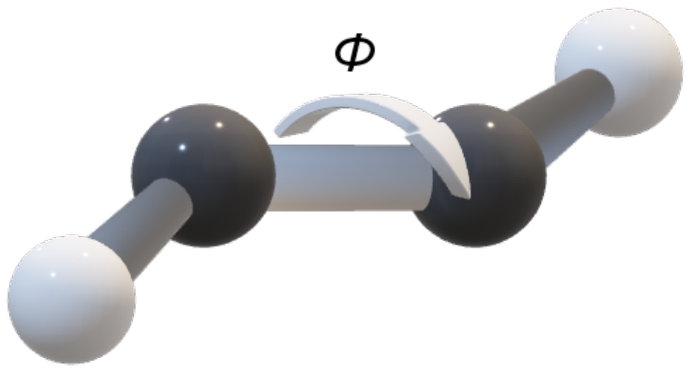
**Quantum mechanics**
very accurate, small system sizes, very expensive to run

# Force Field

$$E_{Total} = \sum_{Bonds} K_r(r - r_0)^2 + \sum_{Angles} K_\theta(\theta - \theta_0)^2 + \sum_{Torsions} \frac{V_n}{2}[1 + cos(n\Phi - \gamma)]$$




**Bonded (Intramolecular) Parameters**

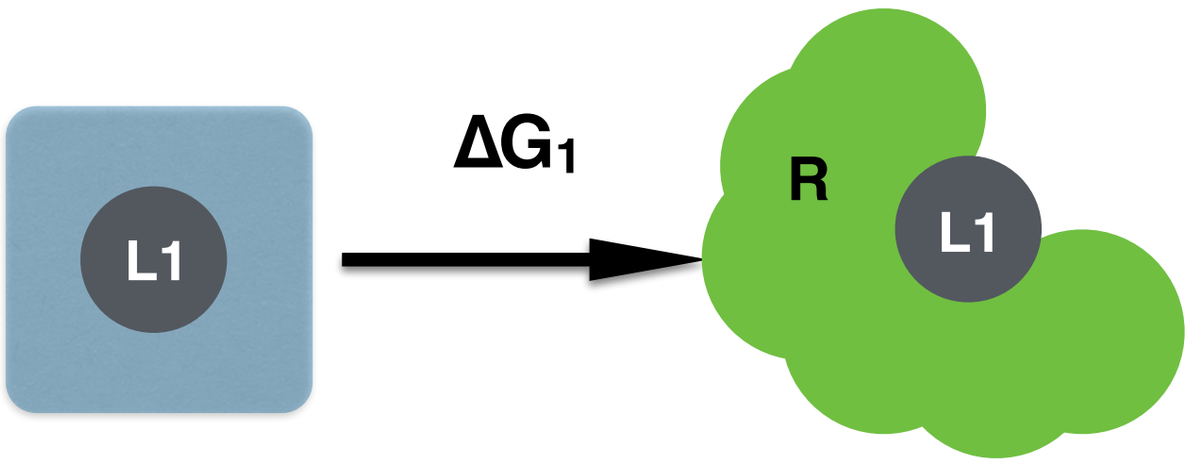
$$+ \sum_{Non-Bonded} \left[ 4\varepsilon_{ij} \left\{ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right\} + \frac{q_i q_j}{r_{ij}} \right]$$


**Non-Bonded (Intermolecular) Parameters**

*An open and collaborative approach to better force fields*

open
forcefield

# Free Energy for Drug Discovery



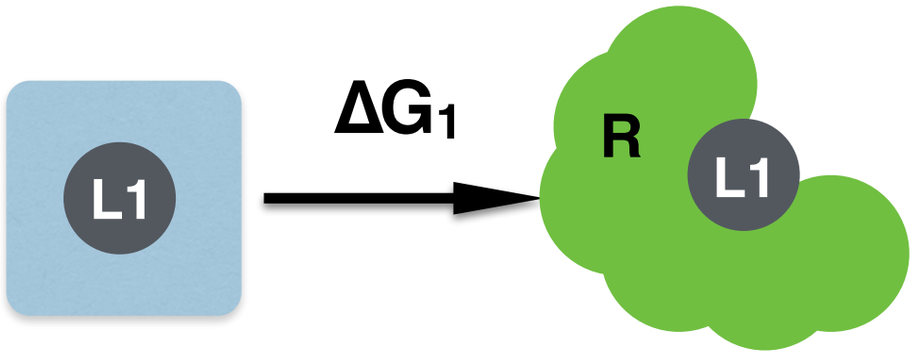


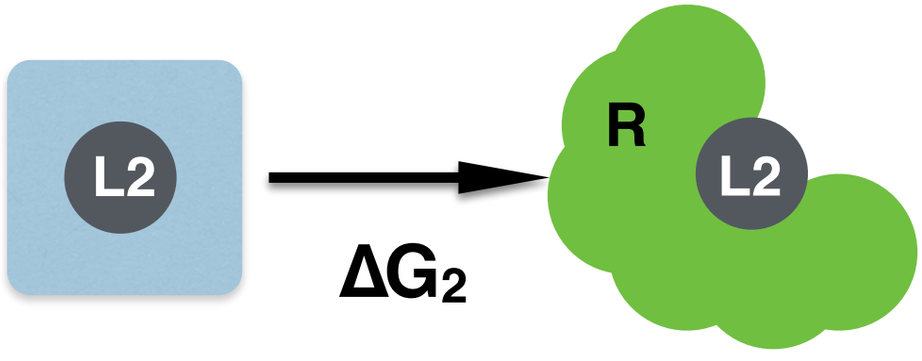
In lead optimisation studies, we are typically interested in optimising the target-ligand binding affinity.

In other words, we need to find the free energy difference between a small molecule (L1) in solution and bound to the protein (R).

Free energy perturbation (FEP) theory provides a rigorous means to compute the binding free energy.

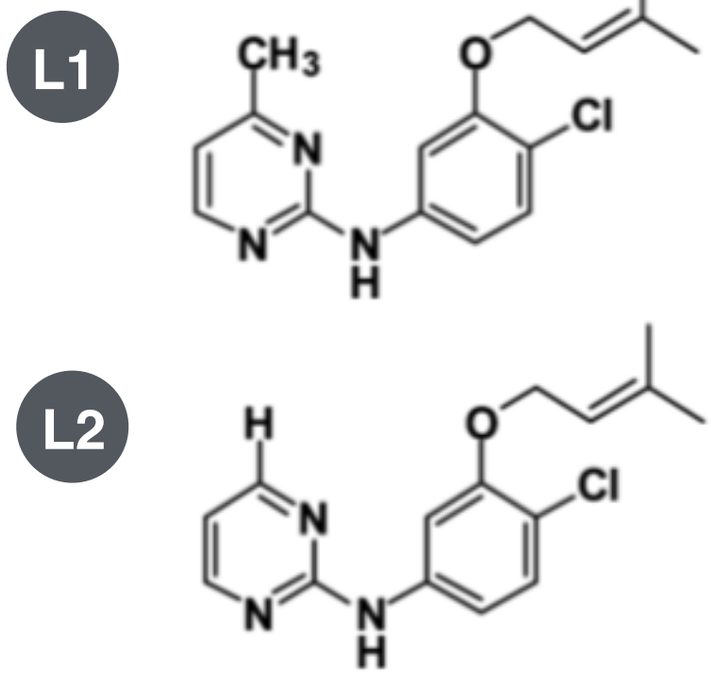
# FEP for Drug Discovery



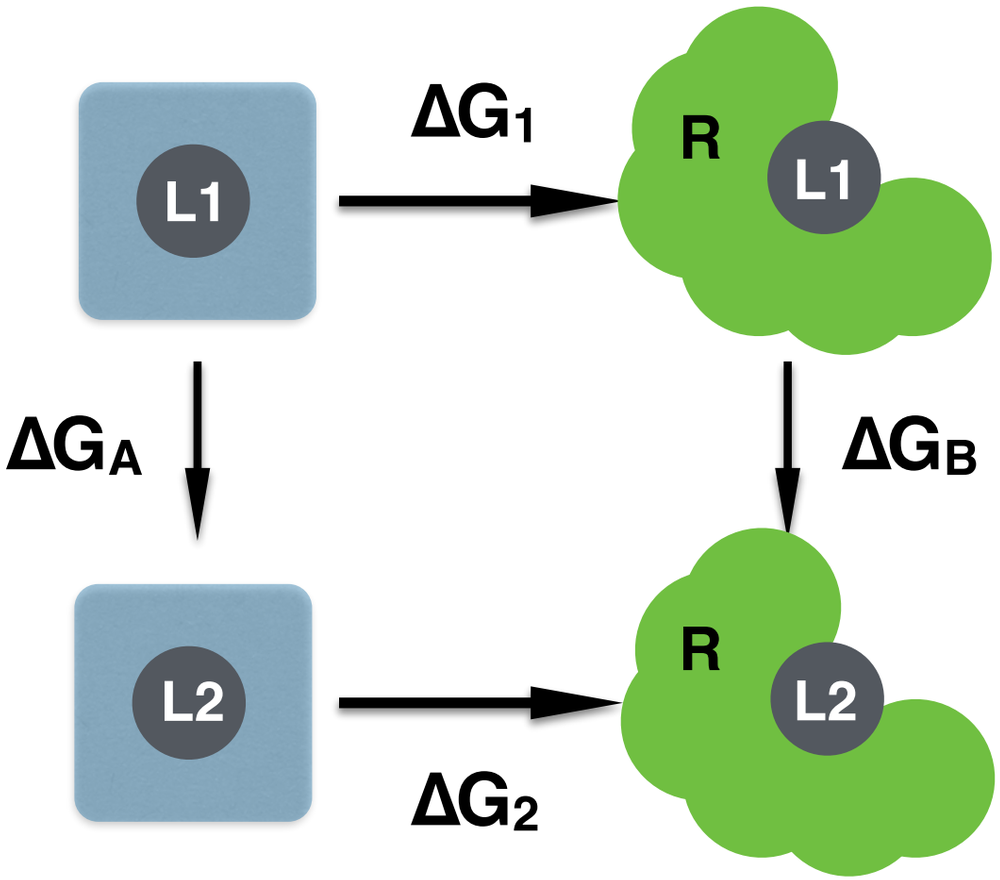

$\Delta\Delta G = \Delta G_2 - \Delta G_1$

For example:




If we have two similar molecules, then often we only need to compute the relative binding free energy ΔΔG.

# FEP for Drug Discovery




The total free energy change around a closed loop is zero:

$$\Delta\Delta G = \Delta G_2 - \Delta G_1 = \Delta G_B - \Delta G_A$$

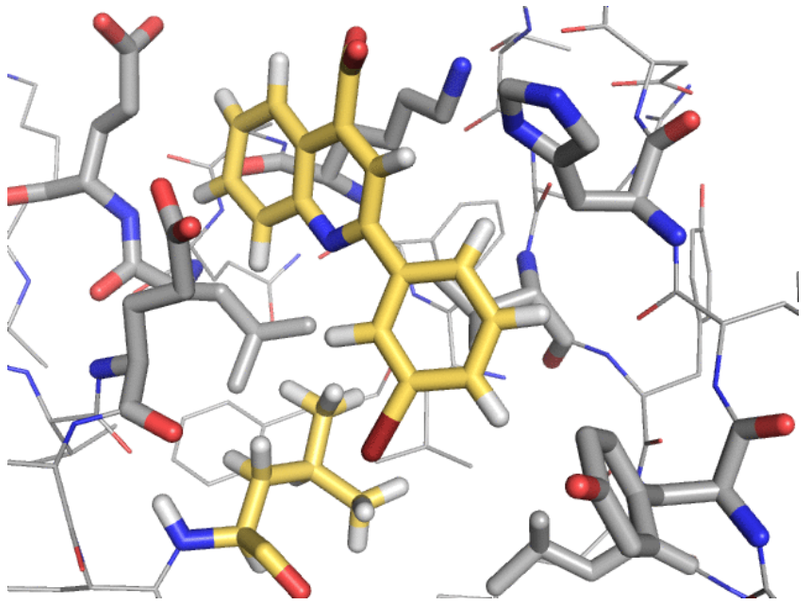
Free energy changes computed using Zwanzig equation:

$$\Delta G_A = -kT\, ln \left\langle exp\left[\frac{-(U_{L2} - U_{L1})}{kT}\right]\right\rangle_{L1}$$

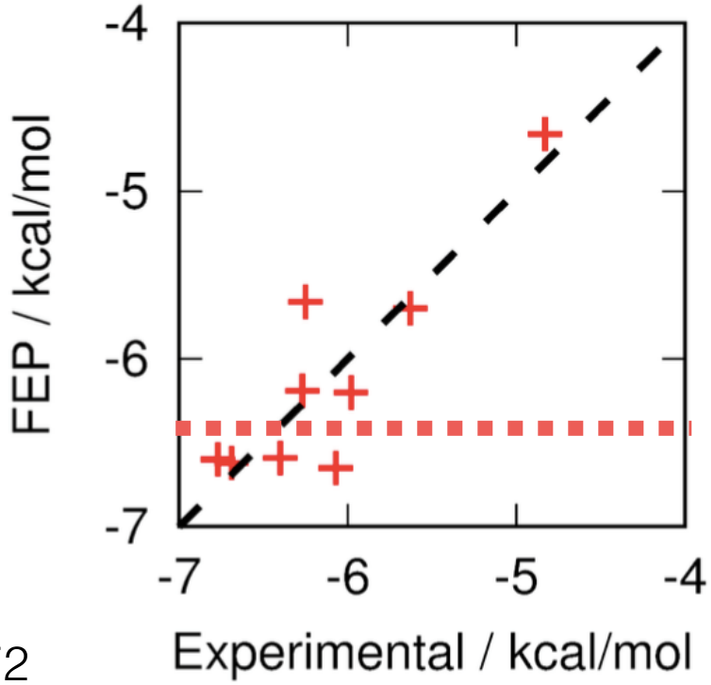
We can use FEP to transform molecule L1 into molecule L2 in the protein and in water. Conformational sampling performed using force field.

# Molecular Interactions & Dynamics

Free energy calculations based on molecular modelling allow us to prioritise molecules for synthesis.



Cole et al., *Chem. Commun.* **2017**, *53*, 9372



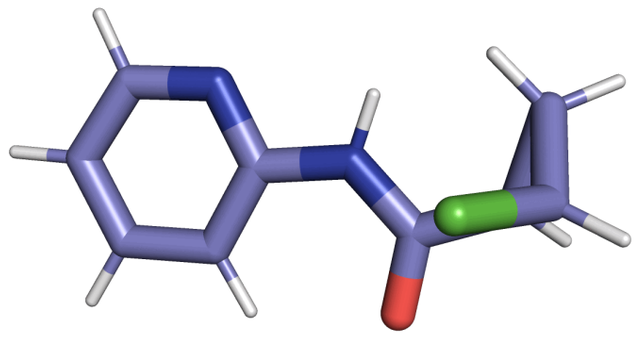

**RMSE =**
**0.32 kcal/mol**

But accuracy is limited by the accuracy of the forces…

# Open Force Field BespokeFit

Accurate determination of molecular conformation is crucial in structure-based drug design.

Molecular conformation is largely determined by torsional rotation about flexible bonds.



open forcefield

Partnered with international industry-academic collaboration to deliver software solution.

**OpenFF-BespokeFit** is an open, automated python package for torsion parameter fitting. Can make use of stored QM data.

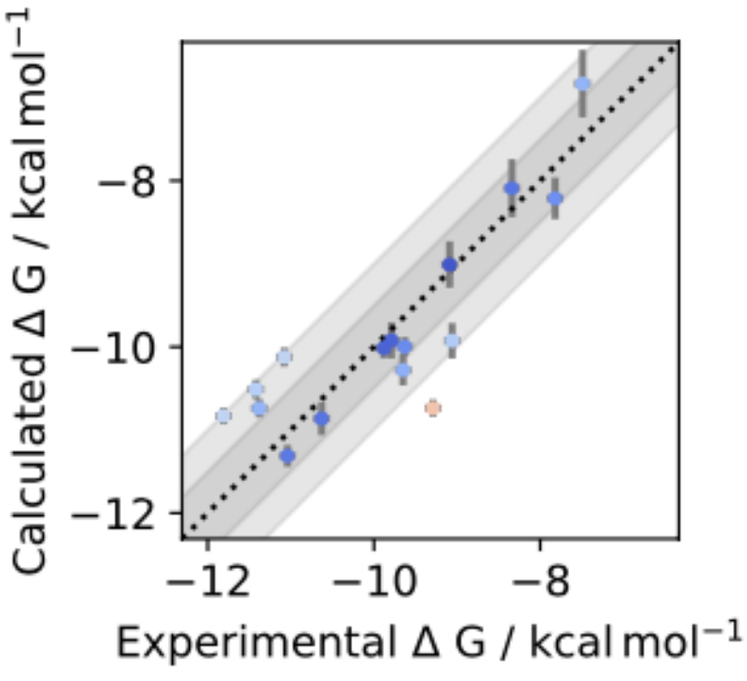

https://github.com/openforcefield/openff-bespokefit

Horton et al. JCIM 2022

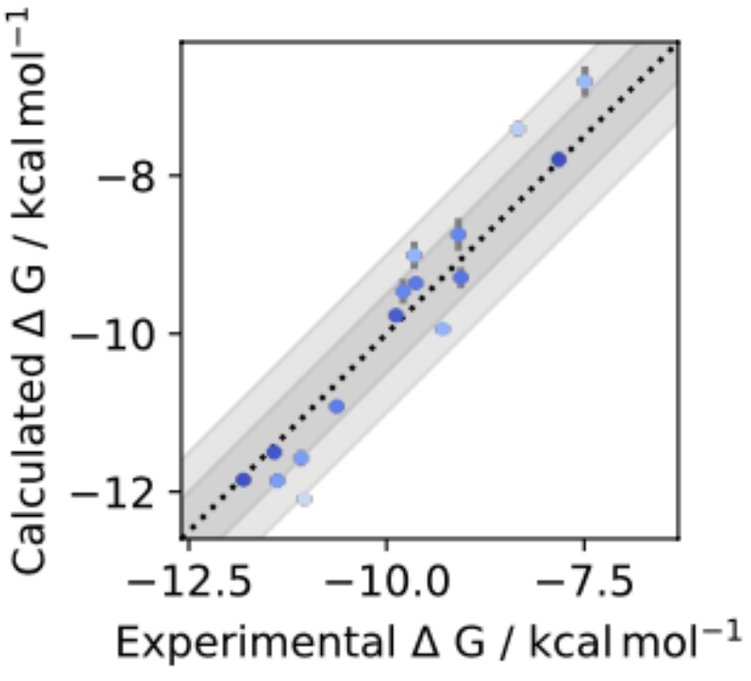
# Bespoke Dihedral Parameters Improve Accuracy

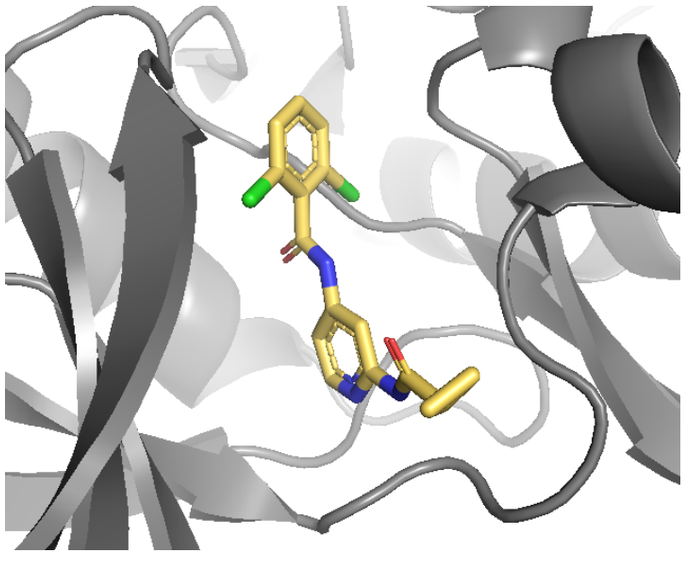
Bespoke torsion parameters improve relative binding free energies for series of TYK2 inhibitors, relative to OpenFF ‘Parsley’ force field.

```
                    OpenFF-1.3
                  tyk2 (N = 16)
RMSE:    0.68 [95%: 0.51, 0.91]
 MUE:    0.56 [95%: 0.39, 0.77]
  R2:    0.72 [95%: 0.35, 0.87]
rho:     0.85 [95%: 0.63, 0.93]
```




```
                  Bespoke default-1.3.0
                        tyk2 (N = 16)
RMSE:    0.51 [95%: 0.35, 0.69]
 MUE:    0.42 [95%: 0.28, 0.59]
  R2:    0.93 [95%: 0.84, 0.97]
rho:     0.97 [95%: 0.92, 0.99]
```

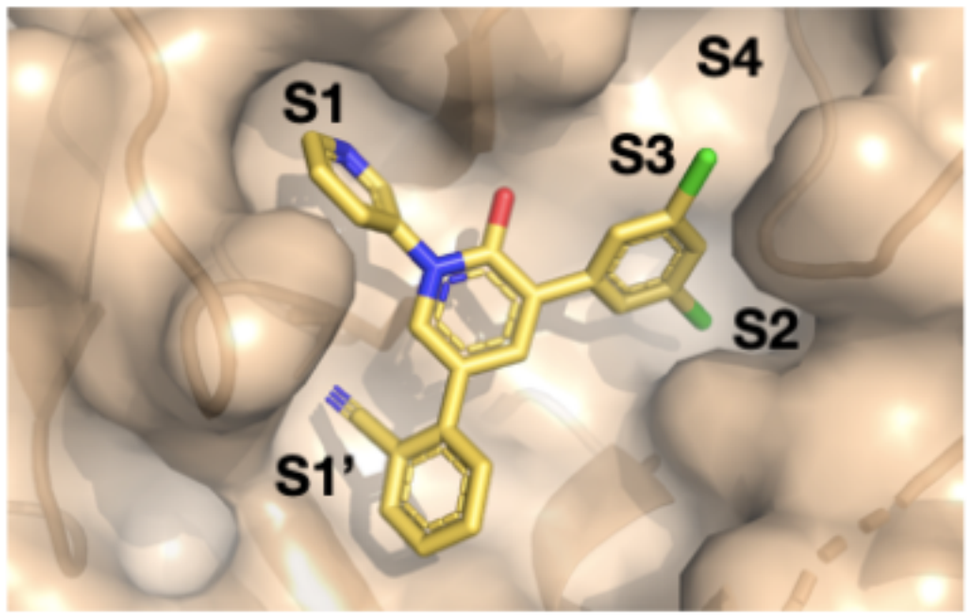


https://github.com/openforcefield/openff-bespokefit

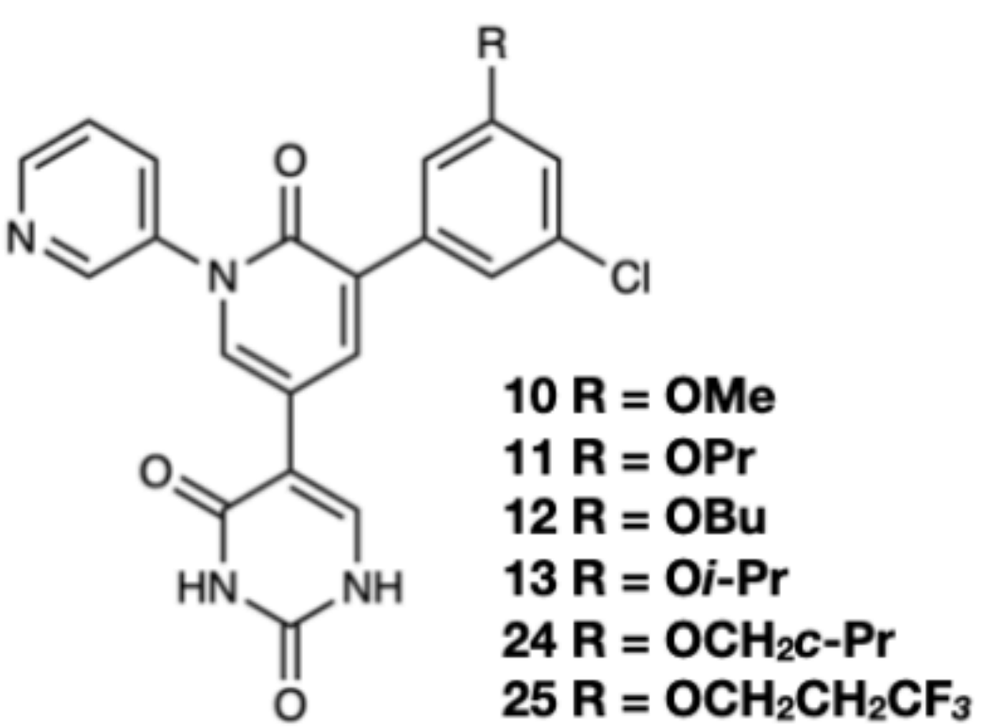
Horton et al. JCIM 2022

# Molecular Design Problem

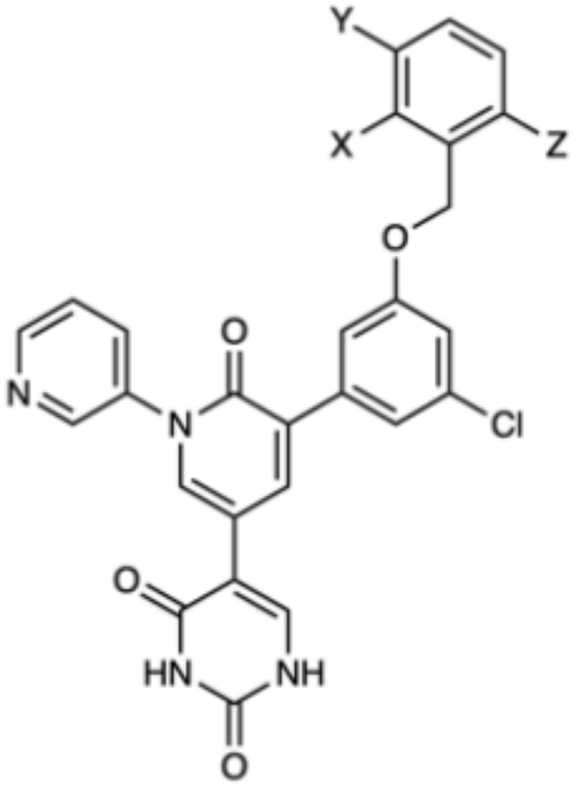



**Typical problem:** design of inhibitors of the main protease of SARS-CoV-2 main protease.

**Lacking open, extensible software** for building and scoring designs.




**10 R = OMe**
**11 R = OPr**
**12 R = OBu**
**13 R = O*i*-Pr**
**24 R = $OCH_2$*c*-Pr**
**25 R = $OCH_2CH_2CF_3$**



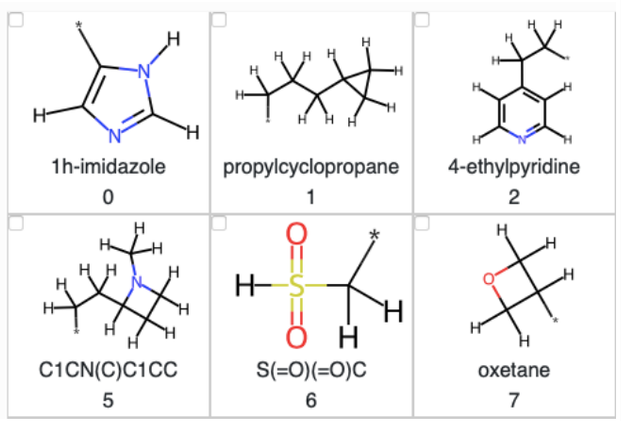

**14 X=Y=Z=H**
**16 X=Me, Y=Z=H**
**17 X=Z=H, Y=Me**
**18 X=F, Y=Z=H**
**19 X=Z=H, Y=F**
**20 X=Z=F, Y=H**
**21 X=Cl, Y=Z=H**

Zhang et al., *ACS Cent. Sci.* **2021**, 7, 467-475

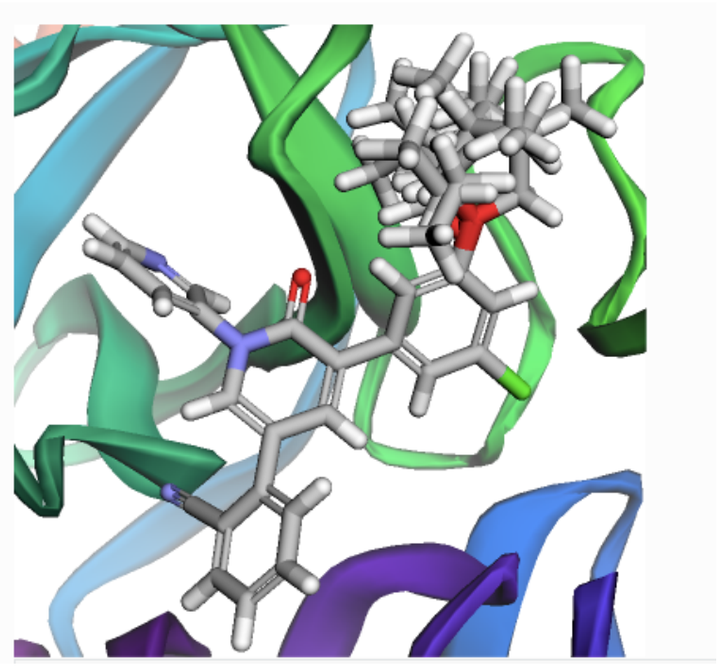
# Our Open Software Solution: FEGrow

An interactive, open source workflow for building congeneric series of ligands.

https://github.com/cole-group/FEgrow



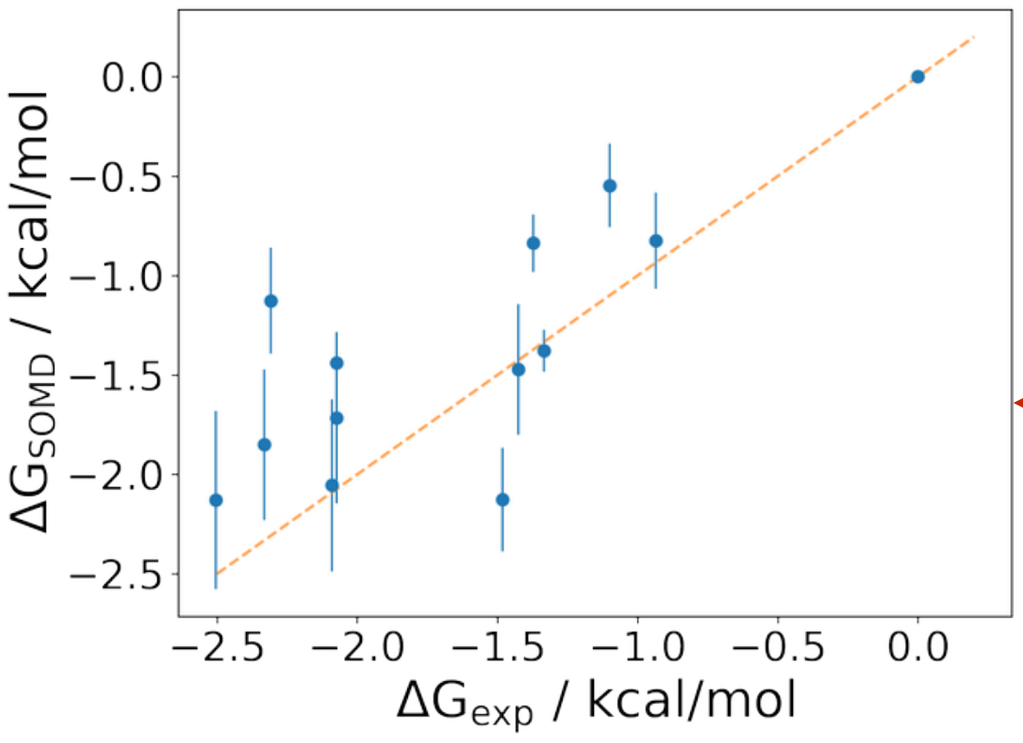

**Build & Enumerate**



**Optimise**



**Score**




M Bieniek, B Cree, R Pirie, J Horton, N Tatum, D Cole, *Commun. Chem.*, **2022**, 5:136
https://github.com/cole-group/FEgrow

# Summary

$$\hat{H}\psi = E\psi$$

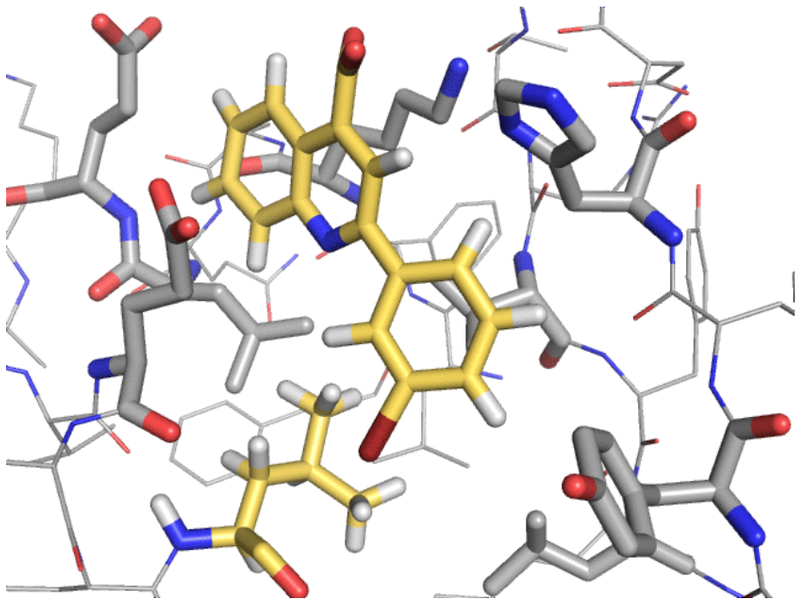
$$\Delta G = -k_B T ln \left\langle exp\left(-\frac{\Delta U}{k_B T}\right)\right\rangle$$

```python
def _calculate_lj_data(self, molecule: "Ligand") -> Dict[int, LJData]:
    """
    Use the AIM parameters to calculate a_i and b_i according to paper.
    Calculations from paper have been combined and simplified for faster computation.
    returns: Dict of the a_i, b_i and r_aim values needed for sigma/epsilon calculation.
    """

    lj_data = {}

    for atom_index, atom in enumerate(molecule.atoms):
        try:
            atomic_symbol, atom_vol = atom.atomic_symbol, atom.aim.volume

            # Find polar Hydrogens and allocate their new name: X
            if atomic_symbol == "H":
                bonded_index = atom.bonds[0]
                if molecule.atoms[bonded_index].atomic_symbol in [
                    "N",
                    "O",
                    "S",
                ]:
                    atomic_symbol = "X"
```



Our goals are to:

1) Develop better approximations to quantum mechanical modelling.

2) Produce software to automate this process.

3) Collect and analyse data to work at scale.

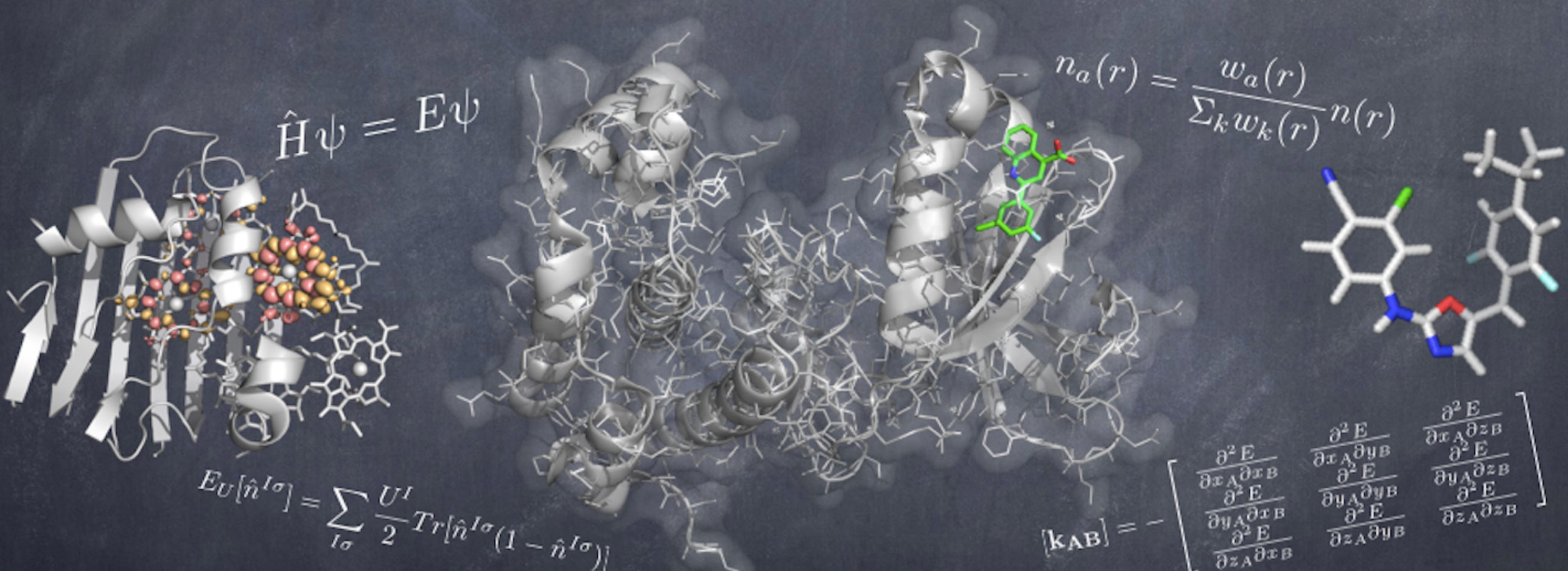
4) Deliver more accurate predictions for drug design.

CCPBioSim

HECBioSim

N8 Bede
COMPUTATIONALLY INTENSIVE RESEARCH

UKRI UK Research and Innovation




**https://blogs.ncl.ac.uk/danielcole/**
**https://github.com/cole-group/**
**@ColeGroupNCL**

Newcastle University

# Thank you for your attention

# Molecular Dynamics

The method we use to **explore** the potential energy surface is called **molecular dynamics (MD)**.

You may also come across Monte Carlo methods. We will not have time to discuss them here.

The basis of molecular dynamics is Newton's second law of motion:

$$\mathbf{F}_i = m_i \mathbf{a}_i = m \frac{d^2 \mathbf{r}_i}{dt^2}$$

If we know the positions of the atoms $\mathbf{r}_i$ at time t=0, then we can find the positions at all later times by integrating the force (from the force field). This gives us a 'movie' of the system.

# Molecular Dynamics

$$\mathbf{F}_i = m_i \mathbf{a}_i = m \frac{d^2 \mathbf{r}_i}{dt^2}$$

If we know the positions of the atoms $\mathbf{r}_i$ at time t=0, then we can find the positions at all later times by integrating the force (from the force field). This gives us a 'movie' of the system.

How do we go about integrating the force with respect to time? We split time up into a series of small, discrete time steps Δt and make use of a Taylor series expansion:

$$f(x + \Delta x) = f(x) + \Delta x \frac{df}{dx} + \frac{1}{2}(\Delta x)^2 \frac{d^2 f}{dx^2} + \ldots$$

# Verlet Algorithm

Applying the Taylor expansion to the positions of the particles:

$$r(t + \Delta t) = r(t) + \Delta t \frac{dr}{dt} + \frac{1}{2}(\Delta t)^2 \frac{d^2 r}{dt^2} + \dots$$

$$r(t - \Delta t) = r(t) - \Delta t \frac{dr}{dt} + \frac{1}{2}(\Delta t)^2 \frac{d^2 r}{dt^2} - \dots$$

Sum the two equations, and rearrange:

$$r(t + \Delta t) = 2r(t) - r(t - \Delta t) + (\Delta t)^2 \frac{d^2 r}{dt^2}$$

$$r(t + \Delta t) = 2r(t) - r(t - \Delta t) + (\Delta t)^2 \frac{F}{m}$$

# Verlet Algorithm

**positions at next time** — **positions at previous times** — **Forces, mass of atoms**

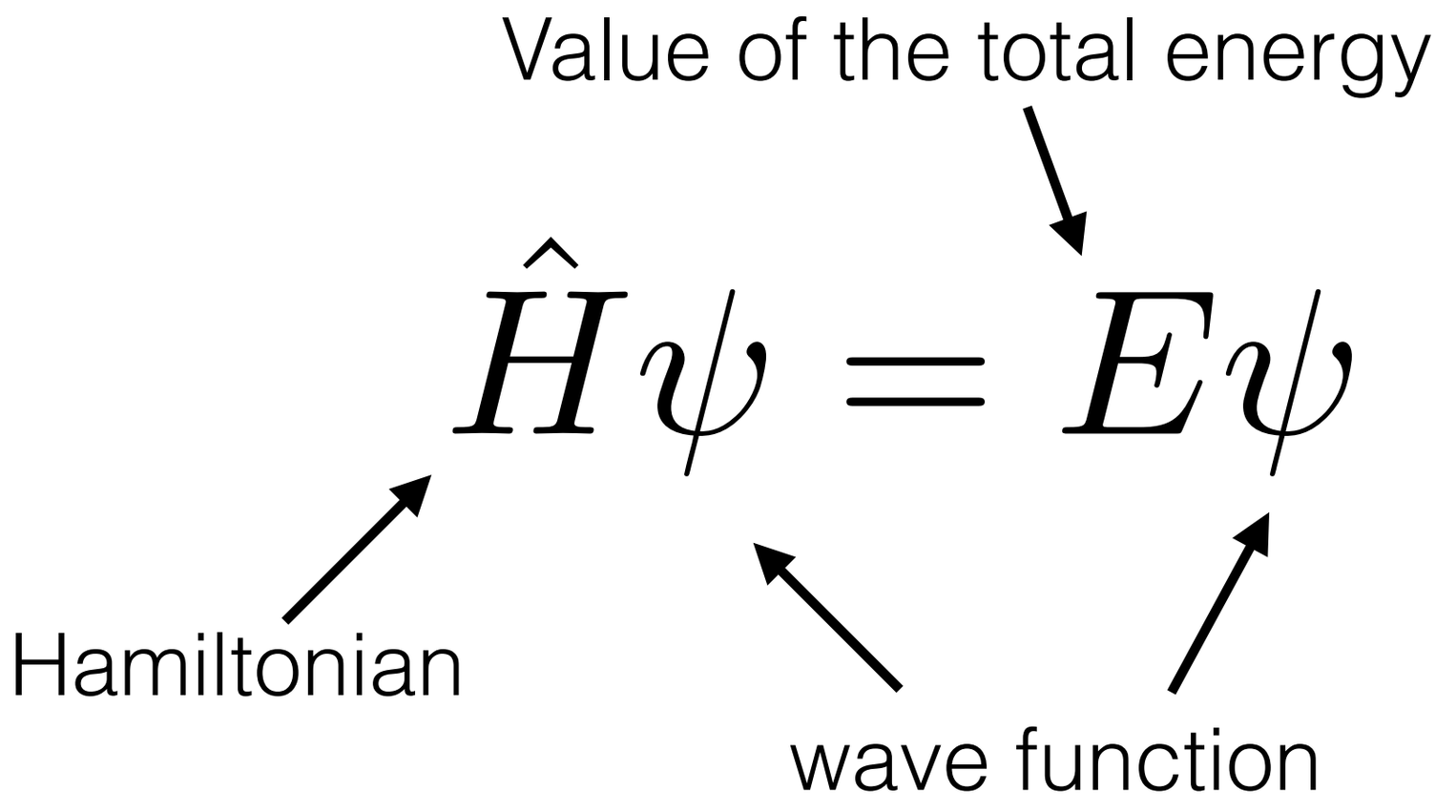
$$r(t+\Delta t) = 2r(t) - r(t-\Delta t) + (\Delta t)^2 \frac{F}{m}$$

This is the **Verlet algorithm**. It allows us to compute atomic positions at all later times, forever, from the initial coordinates and the force field.

The Taylor expansion relies on Δt being small.
How small is small? Given the masses and forces involved, Δt is typically 1 femtosecond ($10^{-15}$ seconds)!

So if we can perform around 100 energy evaluations per second on a cpu, we can simulate around 10 nanoseconds (10ns) per day — not a huge amount but enough to study many important processes.

# Schrödinger Equation




Solution of the Schrödinger equation tells us the **wave function** (where the electrons are) and the **total energy** of a configuration of atoms.

Goal is to find E and ψ, such that action of $\hat{H}$ on ψ returns Eψ.