

Koopmans functionals in practice

minimisation, screening coefficients, and more...

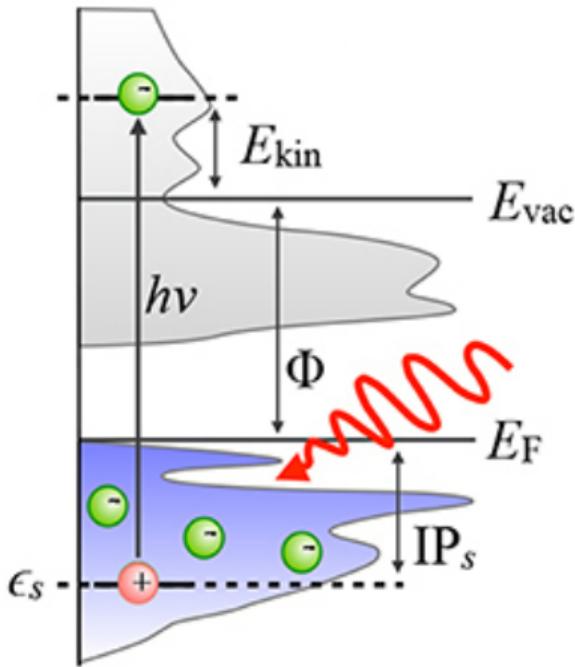
What are Koopmans functionals?

How do they differ from standard DFT?

What implications does this have for running these calculations?

What codes do we use to run these calculations?

Goal: spectral properties (charged excitations) with a functional theory



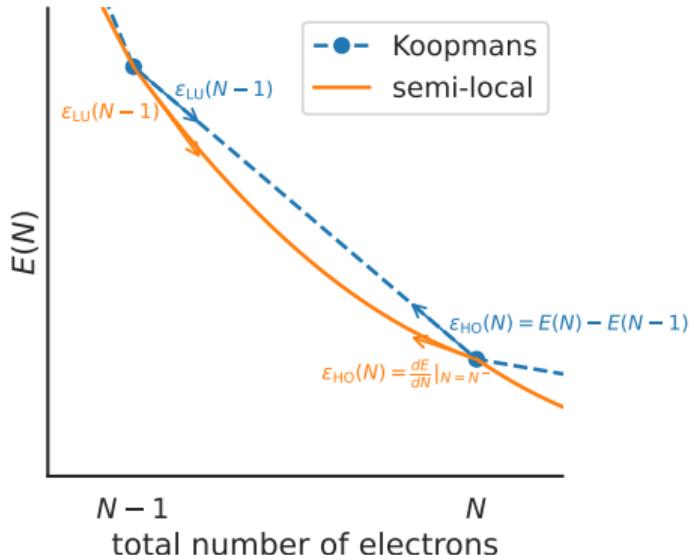
Goal: spectral properties (charged excitations) with a functional theory

Core idea: for every orbital i their energy

$$\varepsilon_i^{\text{Koopmans}} = \langle \varphi_i | H | \varphi_i \rangle = \partial E_{\text{Koopmans}} / \partial f_i$$

should be...

- independent of its own occupation f_i
- equal to the corresponding total energy difference $E_i(N - 1) - E(N)$



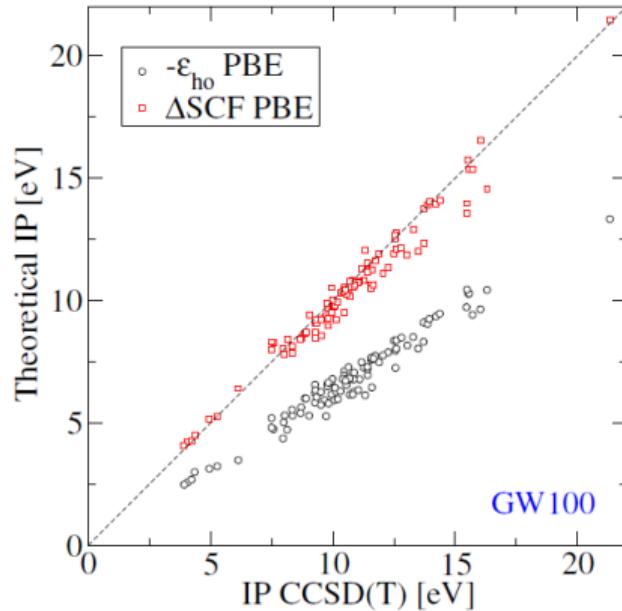
Goal: spectral properties (charged excitations) with a functional theory

Core idea: for every orbital i their energy

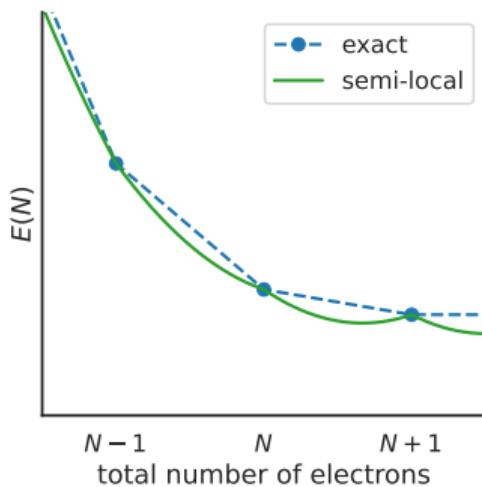
$$\varepsilon_i^{\text{Koopmans}} = \langle \varphi_i | H | \varphi_i \rangle = \partial E_{\text{Koopmans}} / \partial f_i$$

should be...

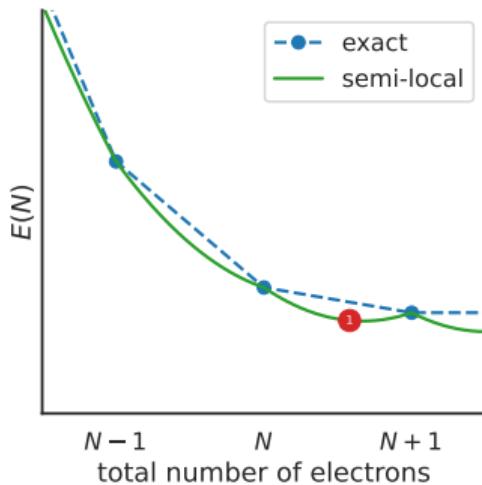
- independent of its own occupation f_i
- equal to the corresponding total energy difference $E_i(N-1) - E(N)$



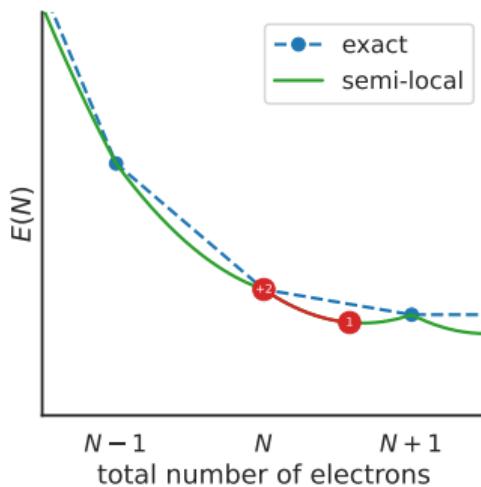
$$-\underbrace{\int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + f_i \underbrace{\int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}}$$



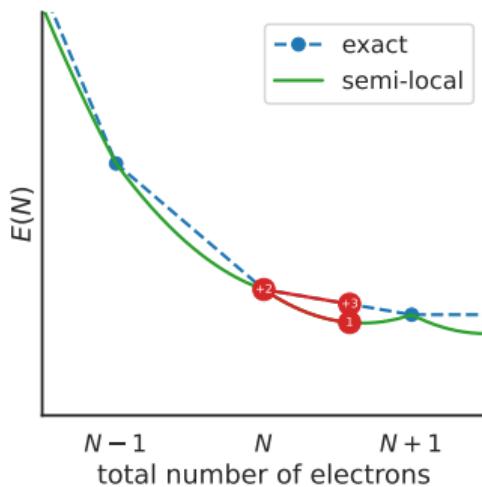
$$-\underbrace{\int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + f_i \underbrace{\int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}}$$



$$-\underbrace{\int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + f_i \underbrace{\int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}}$$



$$-\underbrace{\int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + \underbrace{f_i \int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}}$$



$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left(-\underbrace{\int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + f_i \underbrace{\int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}} \right)$$

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left(-\underbrace{\int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + f_i \underbrace{\int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}} \right)$$

Differences to semi-local functionals:

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left(-\underbrace{\int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + f_i \underbrace{\int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}} \right)$$

Differences to semi-local functionals:

- different flavours

$$E_{\text{Koopmans}}[\rho, \{\mathbf{f}_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left(-\underbrace{\int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + f_i \underbrace{\int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}} \right)$$

Differences to semi-local functionals:

- different flavours
- orbital-density dependence

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left(- \underbrace{\int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + f_i \underbrace{\int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}} \right)$$

Differences to semi-local functionals:

- different flavours
- orbital-density dependence
- screening

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left(- \underbrace{\int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + f_i \underbrace{\int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}} \right)$$

One degree of freedom: what should be the gradient of this linear term?

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left(\underbrace{- \int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + f_i \underbrace{\int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}} \right)$$

One degree of freedom: what should be the gradient of this linear term?

- the base functional → “KI” (Koopmans integral). Does not affect density!

$$E_{\text{Koopmans}}[\rho, \{\mathbf{f}_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left(\underbrace{- \int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + f_i \underbrace{\int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}} \right)$$

One degree of freedom: what should be the gradient of this linear term?

- the base functional → "KI" (Koopmans integral). Does not affect density!
- with a PZ correction → "KIPZ"

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left(\underbrace{- \int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + f_i \underbrace{\int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}} \right)$$

One degree of freedom: what should be the gradient of this linear term?

- the base functional → "KI" (Koopmans integral). Does not affect density!
- with a PZ correction → "KIPZ"

You might also see...

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left(\underbrace{- \int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + f_i \underbrace{\int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}} \right)$$

One degree of freedom: what should be the gradient of this linear term?

- the base functional → “KI” (Koopmans integral). Does not affect density!
- with a PZ correction → “KIPZ”

You might also see...

- “pKIPZ” = KIPZ Hamiltonian evaluated on the KI solution

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left(\underbrace{- \int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + f_i \underbrace{\int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}} \right)$$

One degree of freedom: what should be the gradient of this linear term?

- the base functional → “KI” (Koopmans integral). Does not affect density!
- with a PZ correction → “KIPZ”

You might also see...

- “pKIPZ” = KIPZ Hamiltonian evaluated on the KI solution
- “K” = an earlier iteration based off half-filling rather than integer endpoints (no longer used)

$$-\int_0^{f_i} \varepsilon_i(f) df + f_i \int_0^1 \varepsilon_i(f) df = E_{\text{Hxc}}[\rho] + E_{\text{Hxc}}[\rho - \rho_i] + f_i (-E_{\text{Hxc}}[\rho - \rho_i] + E_{\text{Hxc}}[\rho - \rho_i + n_i])$$

$$-\int_0^{f_i} \varepsilon_i(f) df + f_i \int_0^1 \varepsilon_i(f) df = E_{\text{Hxc}}[\rho] + E_{\text{Hxc}}[\rho - \rho_i] + f_i (-E_{\text{Hxc}}[\rho - \rho_i] + E_{\text{Hxc}}[\rho - \rho_i + n_i])$$

Potential is given by $v_i(\mathbf{r}) = \frac{\delta E}{\delta \rho_i(\mathbf{r})}$. After some derivation...

$$v_i/\alpha_i = v_{\text{scalar}} + \delta_{ij} v_{\text{diag}}(\mathbf{r}) + (1 - \delta_{ij}) v_{\text{nondiag}}(\mathbf{r})$$

$$-\int_0^{f_i} \varepsilon_i(f) df + f_i \int_0^1 \varepsilon_i(f) df = E_{\text{Hxc}}[\rho] + E_{\text{Hxc}}[\rho - \rho_i] + f_i (-E_{\text{Hxc}}[\rho - \rho_i] + E_{\text{Hxc}}[\rho - \rho_i + n_i])$$

Potential is given by $v_i(\mathbf{r}) = \frac{\delta E}{\delta \rho_i(\mathbf{r})}$. After some derivation...

$$v_i/\alpha_i = v_{\text{scalar}} + \delta_{ij} v_{\text{diag}}(\mathbf{r}) + (1 - \delta_{ij}) v_{\text{nondiag}}(\mathbf{r})$$

For filled orbitals with KI:

$$v_i^{\text{KI}}/\alpha_i = -E_{\text{H}}[n_i] + E_{\text{xc}}[\rho] - E_{\text{xc}}[\rho - n_i] - \int d\mathbf{r}' v_{\text{xc}}(\mathbf{r}', [\rho]) n_i(\mathbf{r}')$$

Solve with CG minimisation:

Solve with CG minimisation:

outer loop: $\psi_i^{(n+1)} = \psi_i^{(n)} + \Delta_i$

Solve with CG minimisation:

$$\text{outer loop: } \psi_i^{(n+1)} = \psi_i^{(n)} + \Delta_i$$

$$\text{inner loop: } \psi_i^{(n+1)} = \sum_j U_{ij} \psi_j^{(n)}$$

Solve with CG minimisation:

$$\text{outer loop: } \psi_i^{(n+1)} = \psi_i^{(n)} + \Delta_i$$

$$\text{inner loop: } \psi_i^{(n+1)} = \sum_j U_{ij} \psi_j^{(n)}$$

For more details see G. Borghi et al. *Phys. Rev. B* 91.15 (2015), 155112

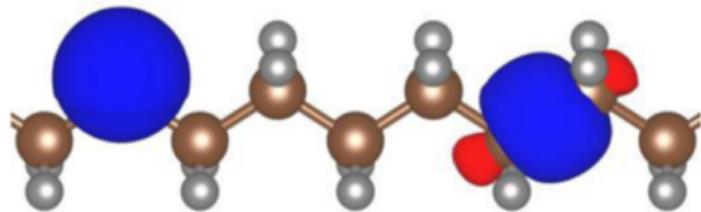
Solve with CG minimisation:

$$\text{outer loop: } \psi_i^{(n+1)} = \psi_i^{(n)} + \Delta_i$$

$$\text{inner loop: } \psi_i^{(n+1)} = \sum_j U_{ij} \psi_j^{(n)}$$

For more details see G. Borghi et al. *Phys. Rev. B* 91.15 (2015), 155112

Gives rise to a set of minimising orbitals (localised/variational)



(a) variational

Orbital-density-dependence

Solve with CG minimisation:

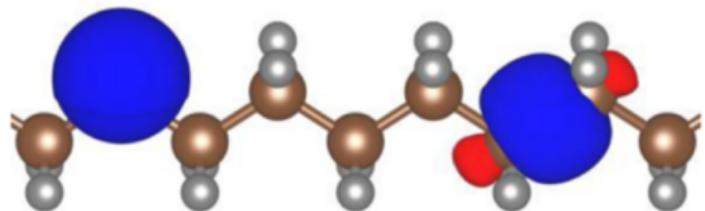
$$\text{outer loop: } \psi_i^{(n+1)} = \psi_i^{(n)} + \Delta_i$$

$$\text{inner loop: } \psi_i^{(n+1)} = \sum_j U_{ij} \psi_j^{(n)}$$

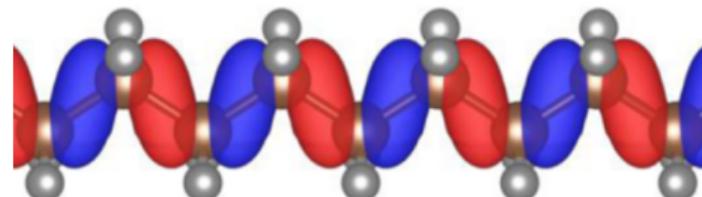
For more details see G. Borghi et al. *Phys. Rev. B* 91.15 (2015), 155112

Gives rise to a set of minimising orbitals (localised/variational)

Diagonalising at the minimum gives rise to diagonalising orbitals (delocalised/canonical)

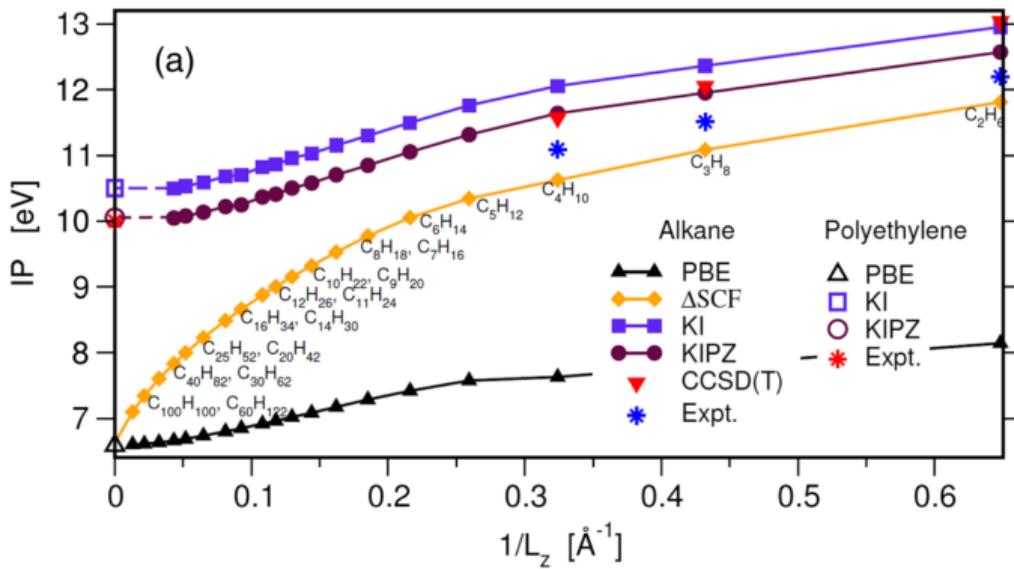


(a) variational

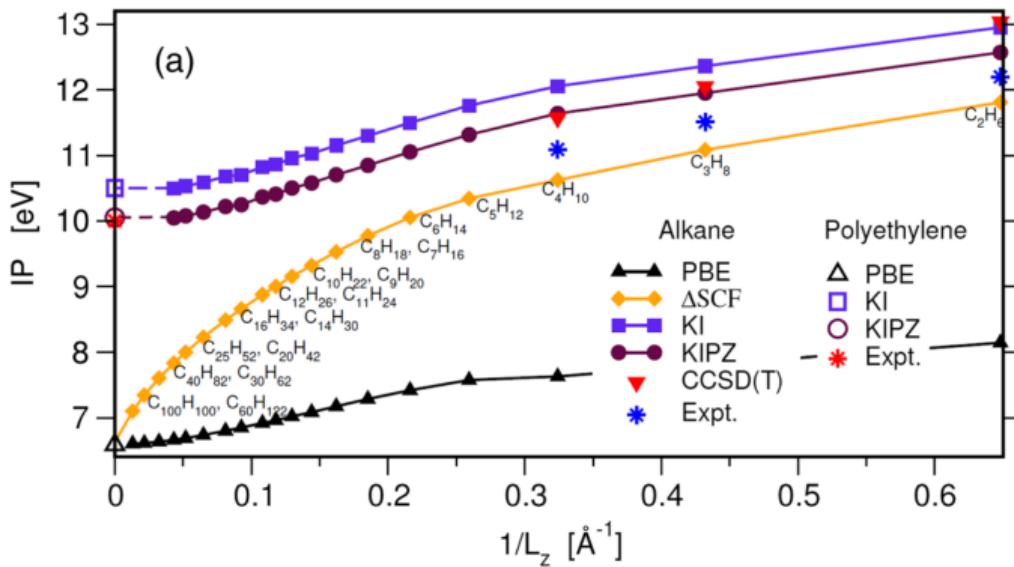


(b) canonical

Importance of localisation

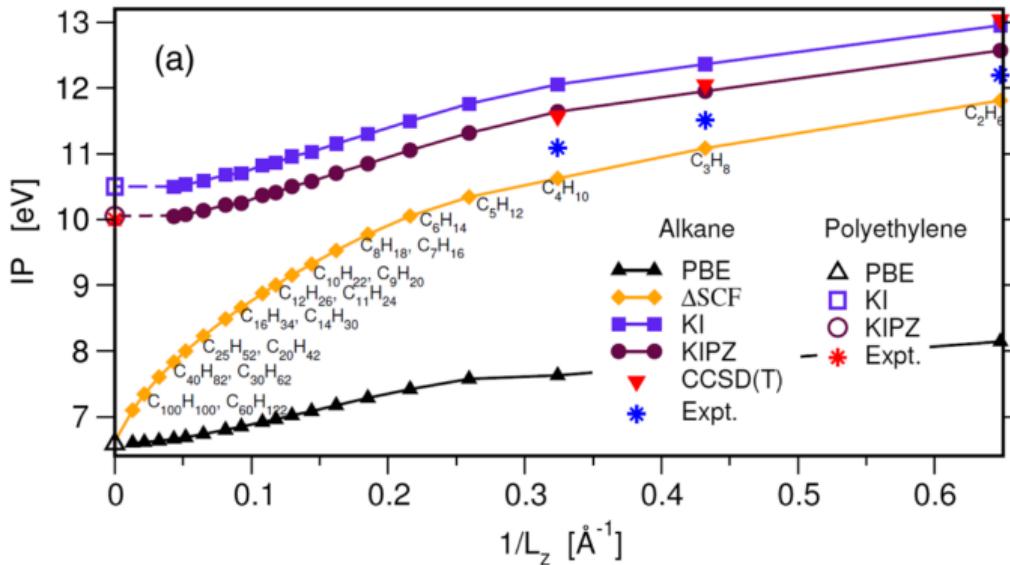


Importance of localisation



In the bulk limit for one cell $\Delta E = E(N - \delta) - E(N)$

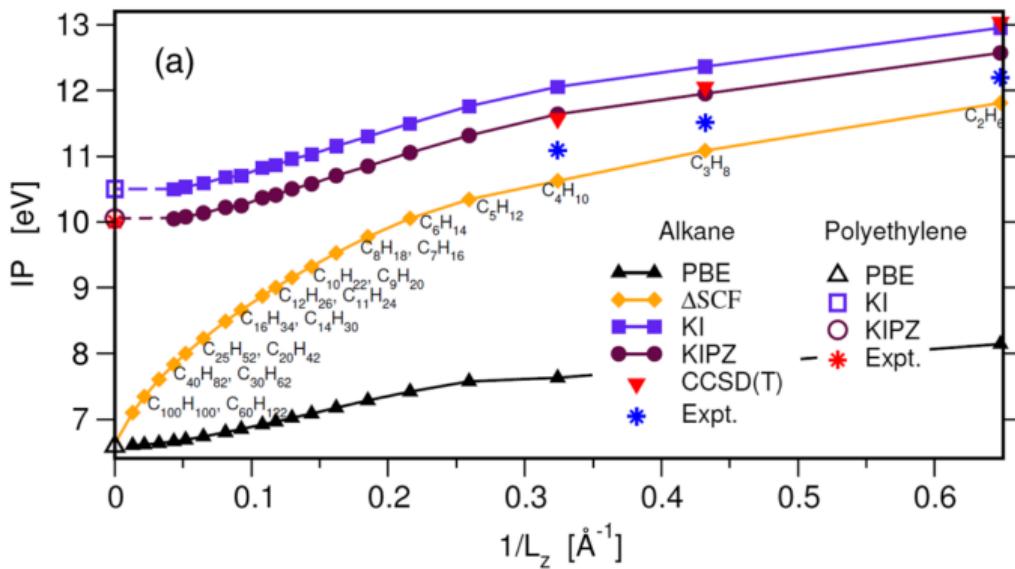
Importance of localisation



In the bulk limit for one cell $\Delta E = E(N - \delta) - E(N)$

Across all the cells $\Delta E = \frac{1}{\delta N} (E(N - \delta N) - E(N)) = -\frac{dE}{dN} = -\varepsilon_{HO}$

Importance of localisation



$$\lim_{n_i(r) \rightarrow 0} v_i^{\text{KI}} / \alpha_i = \lim_{n_i(r) \rightarrow 0} -E_H[n_i] + E_{\text{xc}}[\rho] - E_{\text{xc}}[\rho - n_i] - \int d\mathbf{r}' v_{\text{xc}}(\mathbf{r}', [\rho]) n_i(\mathbf{r}') = 0$$

Other features of orbital-density-dependence

- ODD functional means that we know $\hat{H}|\varphi_i\rangle$ for variational orbitals $\{|\varphi_i\rangle\}$ but we don't know \hat{H} in general

¹ A. Ferretti et al. *Phys. Rev. B* 89.19 (2014), 195134.

Other features of orbital-density-dependence

- ODD functional means that we know $\hat{H}|\varphi_i\rangle$ for variational orbitals $\{|\varphi_i\rangle\}$ but we don't know \hat{H} in general
- Practically we can often use MLWFs

¹ A. Ferretti et al. *Phys. Rev. B* 89.19 (2014), 195134.

Other features of orbital-density-dependence

- ODD functional means that we know $\hat{H}|\varphi_i\rangle$ for variational orbitals $\{|\varphi_i\rangle\}$ but we don't know \hat{H} in general
- Practically we can often use MLWFs
- a natural generalisation in the direction of spectral functional theory¹

¹ A. Ferretti et al. *Phys. Rev. B* 89.19 (2014), 195134.

- What is screening (in the context of Koopmans functionals)?

$$\underbrace{\frac{dE}{df_i}}_{\text{including relaxation}} \approx \alpha_i \underbrace{\frac{\partial E}{\partial f_i}}_{\text{excluding relaxation}}$$

- What is screening (in the context of Koopmans functionals)?

$$\underbrace{\frac{dE}{df_i}}_{\text{including relaxation}} \approx \alpha_i \underbrace{\frac{\partial E}{\partial f_i}}_{\text{excluding relaxation}}$$

- we see this in Hartree-Fock (the original “Koopmans’ theorem”):

$$E_{ee}^{HF} = \frac{1}{2} \sum_{ij} f_i f_j \int d\mathbf{r} d\mathbf{r}' \frac{|\psi_i(\mathbf{r})|^2 |\psi_j(\mathbf{r}')|^2}{\mathbf{r} - \mathbf{r}'} - \frac{\psi_i^*(\mathbf{r}) \psi_j^*(\mathbf{r}') \psi_i(\mathbf{r}') \psi_j(\mathbf{r})}{\mathbf{r} - \mathbf{r}'}$$

- What is screening (in the context of Koopmans functionals)?

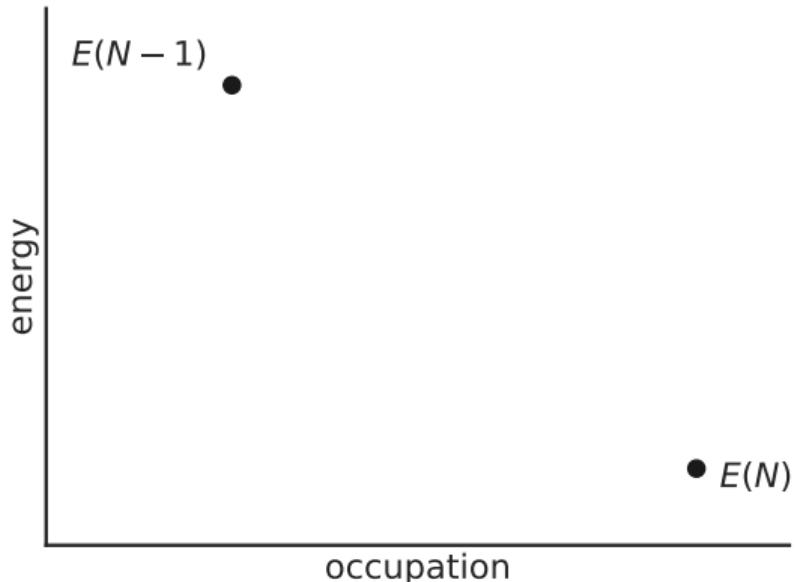
$$\underbrace{\frac{dE}{df_i}}_{\text{including relaxation}} \approx \alpha_i \underbrace{\frac{\partial E}{\partial f_i}}_{\text{excluding relaxation}}$$

- we see this in Hartree-Fock (the original “Koopmans’ theorem”):

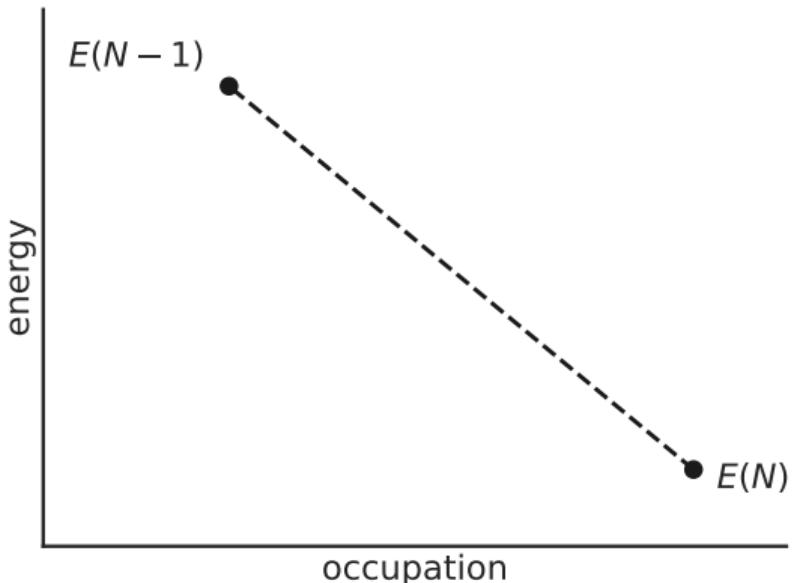
$$E_{ee}^{HF} = \frac{1}{2} \sum_{ij} f_i f_j \int d\mathbf{r} d\mathbf{r}' \frac{|\psi_i(\mathbf{r})|^2 |\psi_j(\mathbf{r}')|^2}{\mathbf{r} - \mathbf{r}'} - \frac{\psi_i^*(\mathbf{r}) \psi_j^*(\mathbf{r}') \psi_i(\mathbf{r}') \psi_j(\mathbf{r})}{\mathbf{r} - \mathbf{r}'}$$

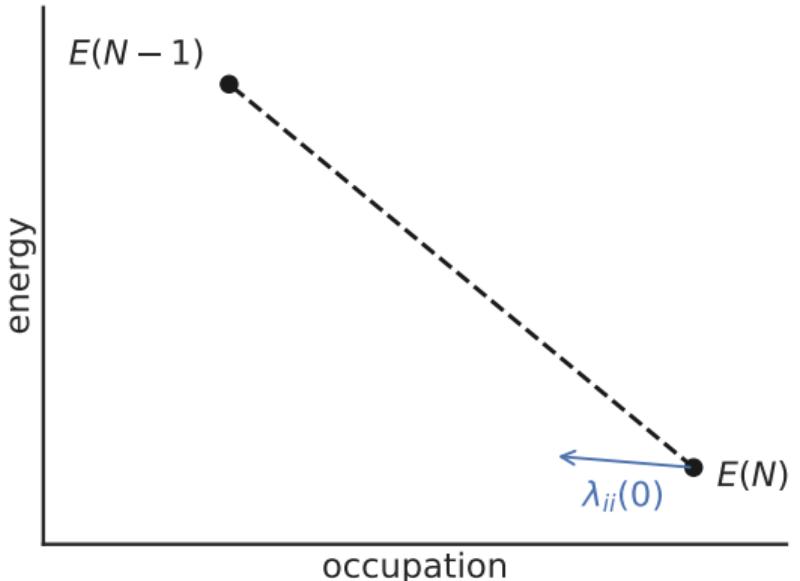
- How to choose an appropriate value for α_i ? Return to the original idea of Koopmans functionals:

$$\varepsilon_i^{\text{Koopmans}} = E_i(N-1) - E(N)$$

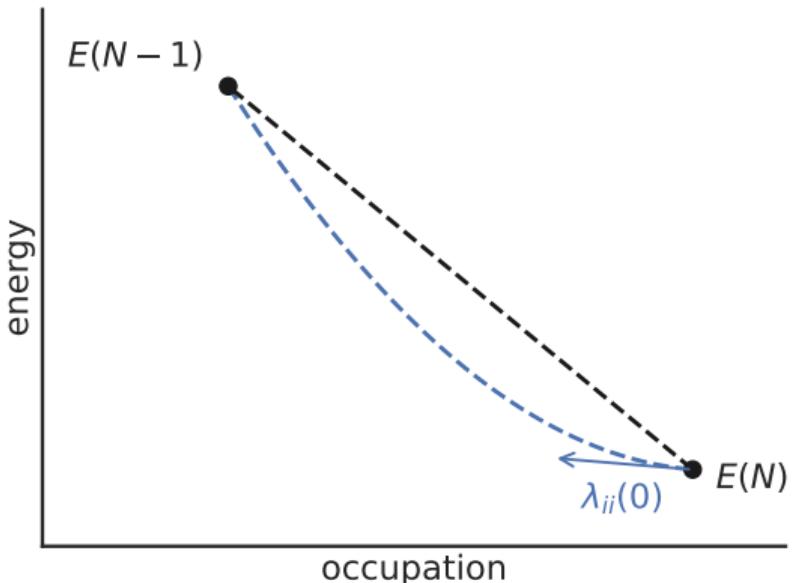


Screening

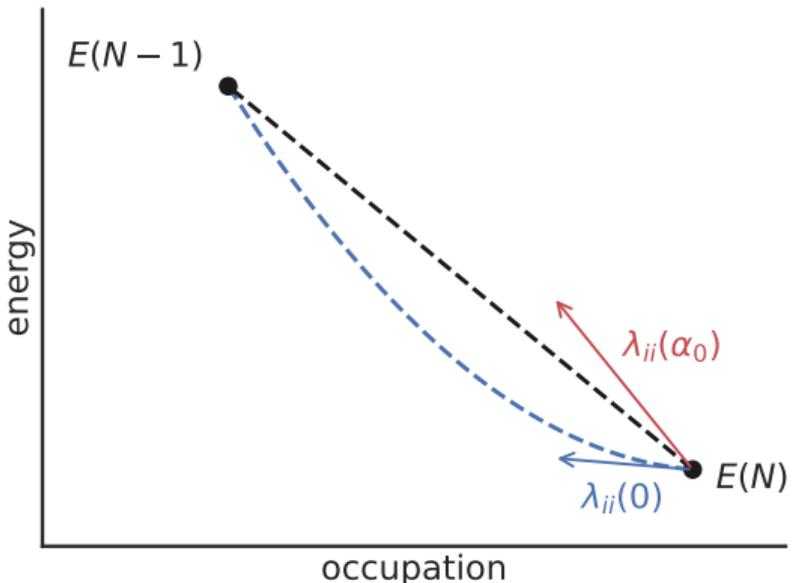




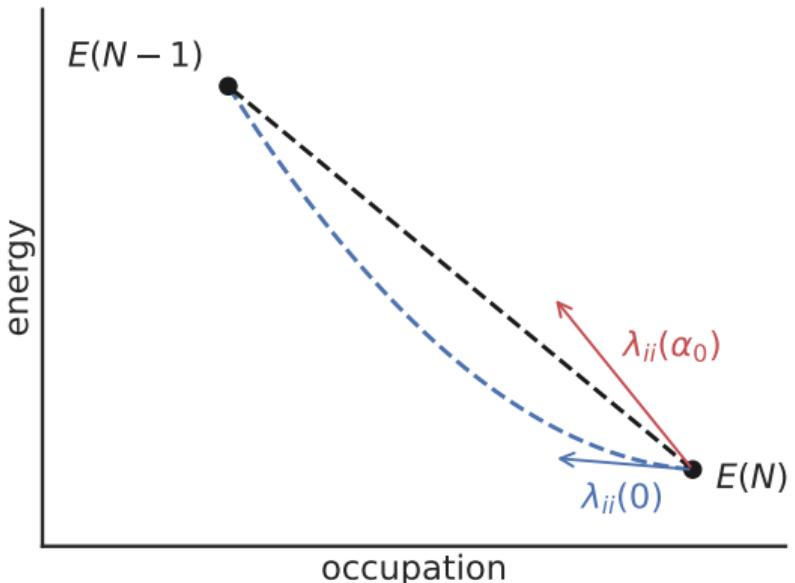
$$\lambda_{ii}(\alpha) \equiv \langle \varphi_i | \hat{h}^{\text{DFT}} + \alpha_i \hat{v}^{\text{Koopmans}} | \varphi_i \rangle = \frac{dE^{\text{Koopmans}}}{df_i} \Big|_{f_i=s}$$



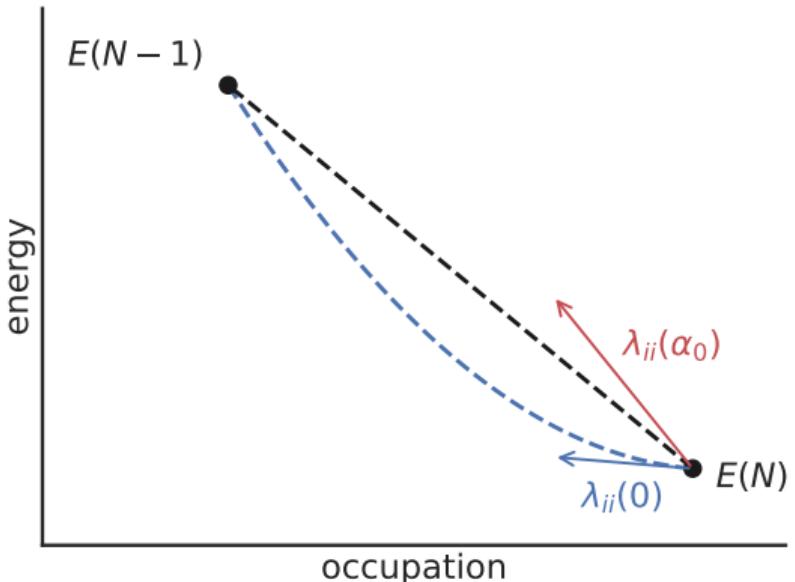
$$\lambda_{ii}(\alpha) \equiv \langle \varphi_i | \hat{h}^{\text{DFT}} + \alpha_i \hat{v}^{\text{Koopmans}} | \varphi_i \rangle = \frac{dE^{\text{Koopmans}}}{df_i} \Big|_{f_i=s}$$



$$\lambda_{ii}(\alpha) \equiv \langle \varphi_i | \hat{h}^{\text{DFT}} + \alpha_i \hat{v}^{\text{Koopmans}} | \varphi_i \rangle = \frac{dE^{\text{Koopmans}}}{df_i} \Big|_{f_i=s}$$



Given this, how to work out the ideal α ?



Given this, how to work out the ideal α ?

$$\alpha^{n+1} = \alpha^n \frac{E_i(N-1) - E(N) - \lambda_{ii}(0)}{\lambda_{ii}(\alpha^n) - \lambda_{ii}(0)}$$

$$\alpha^{n+1} = \alpha^n \frac{E_i(N-1) - E(N) - \lambda_{ii}(0)}{\lambda_{ii}(\alpha^n) - \lambda_{ii}(0)}$$

$$\alpha^{n+1} = \alpha^n \frac{E_i(N-1) - E(N) - \lambda_{ii}(0)}{\lambda_{ii}(\alpha^n) - \lambda_{ii}(0)}$$

↑
expectation value
of $\hat{H}^{\text{Koopmans}}$

$$\alpha^{n+1} = \alpha^n \frac{E_i(N-1) - E(N) - \lambda_{ii}(0)}{\lambda_{ii}(\alpha^n) - \lambda_{ii}(0)}$$

↑
expectation value
of \hat{H}^{DFT}
↑
expectation value
of $\hat{H}^{\text{Koopmans}}$

$$\alpha^{n+1} = \alpha^n \frac{E_i(N-1) - E(N) - \lambda_{ii}(0)}{\lambda_{ii}(\alpha^n) - \lambda_{ii}(0)}$$

total energy
of neutral system

↑
↑
↑

expectation value
of \hat{H}^{DFT}

expectation value
of $\hat{H}^{\text{Koopmans}}$

$$\alpha^{n+1} = \alpha^n \frac{E_i(N-1) - E(N) - \lambda_{ii}(0)}{\lambda_{ii}(\alpha^n) - \lambda_{ii}(0)}$$

total energy with electron removed from orbital i

total energy of neutral system

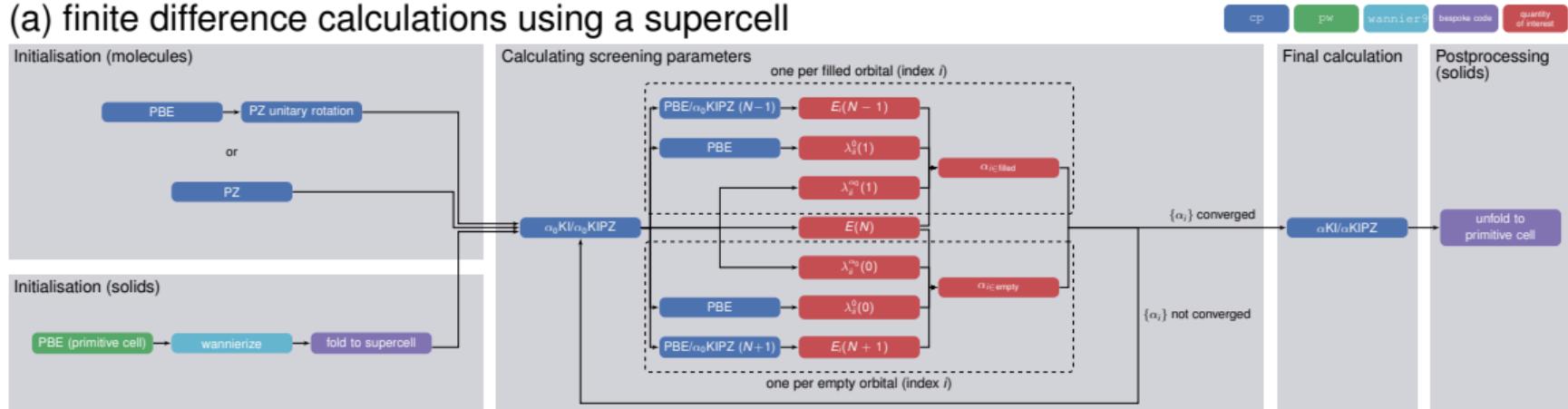
expectation value of \hat{H}^{DFT}

expectation value of $\hat{H}^{\text{Koopmans}}$

Workflows for calculating screening parameters

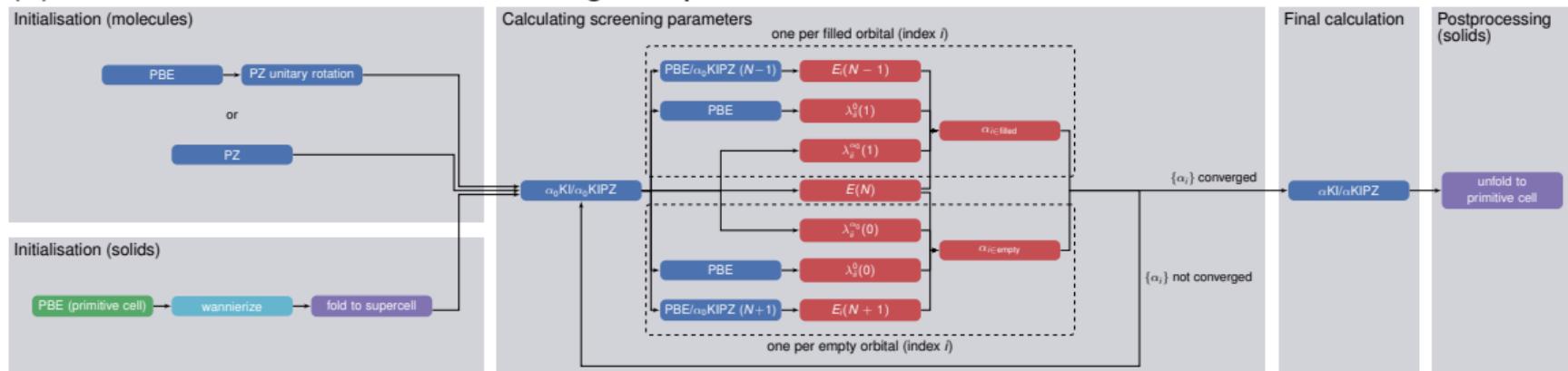
Workflows for calculating screening parameters

(a) finite difference calculations using a supercell



Workflows for calculating screening parameters

(a) finite difference calculations using a supercell

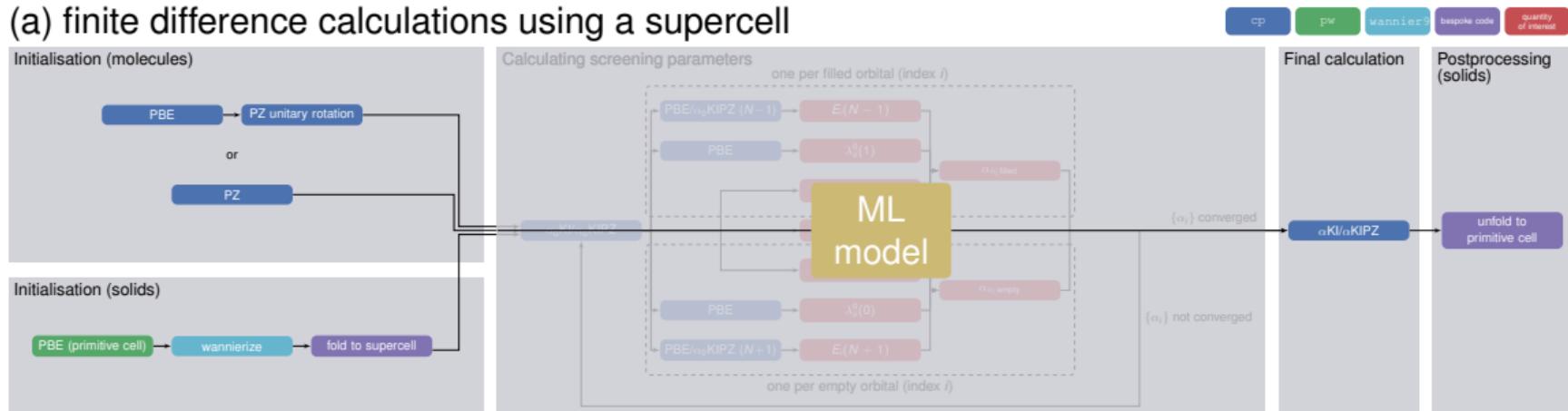


(b) DFPT (see Nicola Colonna's talk)

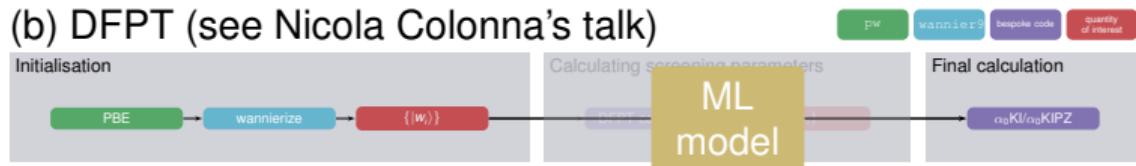


Workflows for calculating screening parameters

(a) finite difference calculations using a supercell



(b) DFPT (see Nicola Colonna's talk)



(c) via machine learning (see Yannick Schubert's talk)

- Wannier transition-state method of Anisimov and Kozhevnikov V. I. Anisimov et al. *Phys. Rev. B* 72.7 (2005), 075125
- Optimally tuned hybrid functionals of Kronik, Pasquarello, and others (refer back to Leeor's talk on Wednesday) L. Kronik et al. *J. Chem. Theory Comput.* 8.5 (2012), 1515; D. Wing et al. *Proc. Natl. Acad. Sci.* 118.34 (2021), e2104556118
- Ensemble DFT of Kronik and co-workers E. Kraisler et al. *Phys. Rev. Lett.* 110.12 (2013), 126403
- Koopmans-Wannier of Wang and co-workers J. Ma et al. *Sci. Rep.* 6.1 (2016), 24924
- Dielectric-dependent hybrid functionals of Galli and co-workers J. H. Skone et al. *Phys. Rev. B* 93.23 (2016), 235106
- LOSC functionals of Yang and co-workers C. Li et al. *Natl. Sci. Rev.* 5 (2018), 203

	DFT+ U	Koopmans
the functional...	corrects local curvature in total energies	
correction applied to...	selected subspaces only (e.g. 3d orbitals)	
orbitals defined by...	Hubbard projectors (atom-centred, frozen, incomplete)	
corrective parameters are...	$\{U^I\}$, defined with respect to charge-neutral excitations (if using LR)	

	DFT+ <i>U</i>	Koopmans
the functional...	corrects local curvature in total energies	removes dependence of ε_i on f_i and guarantees $\varepsilon_i = E_i(N \pm 1) - E(N)$
correction applied to...	selected subspaces only (e.g. 3d orbitals)	
orbitals defined by...	Hubbard projectors (atom-centred, frozen, incomplete)	
corrective parameters are...	$\{U^I\}$, defined with respect to charge-neutral excitations (if using LR)	

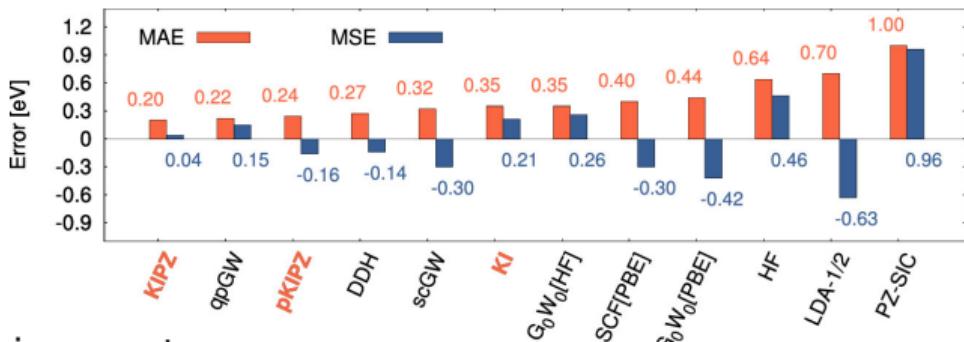
	DFT+ <i>U</i>	Koopmans
the functional...	corrects local curvature in total energies	removes dependence of ε_i on f_i and guarantees $\varepsilon_i = E_i(N \pm 1) - E(N)$
correction applied to...	selected subspaces only (e.g. 3d orbitals)	every variational orbital in the entire system
orbitals defined by...	Hubbard projectors (atom-centred, frozen, incomplete)	
corrective parameters are...	$\{U'\}$, defined with respect to charge-neutral excitations (if using LR)	

	DFT+ U	Koopmans
the functional...	corrects local curvature in total energies	removes dependence of ε_i on f_i and guarantees $\varepsilon_i = E_i(N \pm 1) - E(N)$
correction applied to...	selected subspaces only (e.g. 3d orbitals)	every variational orbital in the entire system
orbitals defined by...	Hubbard projectors (atom-centred, frozen, incomplete)	variational (minimising) orbitals
corrective parameters are...	$\{U'\}$, defined with respect to charge-neutral excitations (if using LR)	

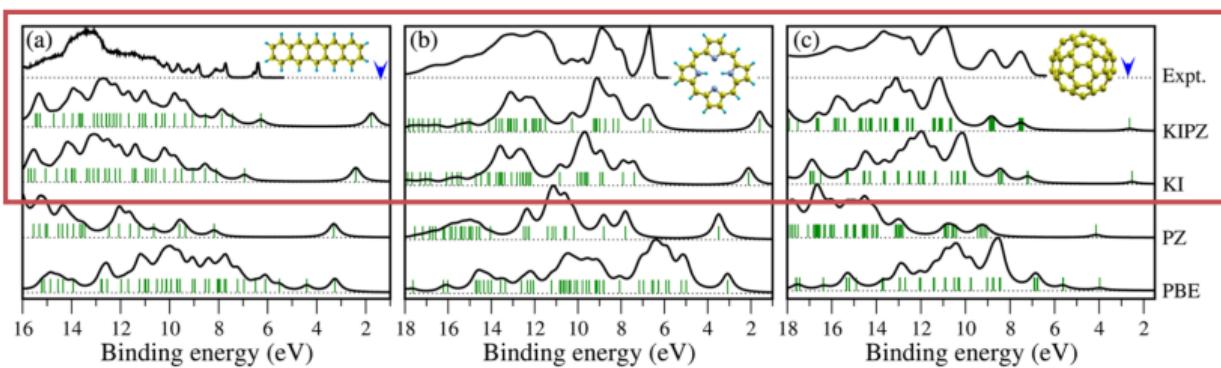
	DFT+ U	Koopmans
the functional...	corrects local curvature in total energies	removes dependence of ε_i on f_i and guarantees $\varepsilon_i = E_i(N \pm 1) - E(N)$
correction applied to...	selected subspaces only (e.g. 3d orbitals)	every variational orbital in the entire system
orbitals defined by...	Hubbard projectors (atom-centred, frozen, incomplete)	variational (minimising) orbitals
corrective parameters are...	$\{U^I\}$, defined with respect to charge-neutral excitations (if using LR)	$\{\alpha_i\}$, defined with respect to charged excitations

Koopmans functionals: results for molecules

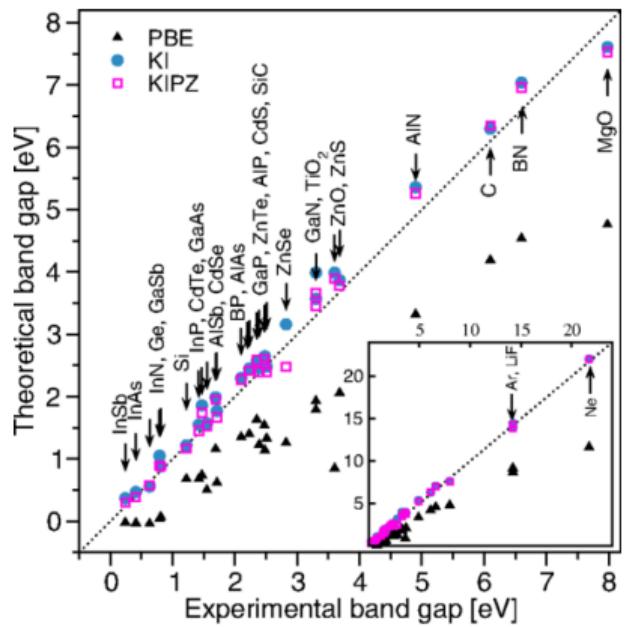
Ionisation potentials = $E(N - 1) - E(N) \stackrel{?}{=} -\varepsilon_{HO}$ of 100 molecules (the GW100 set) cf. CCSD(T)



Ultraviolet photoemission spectra



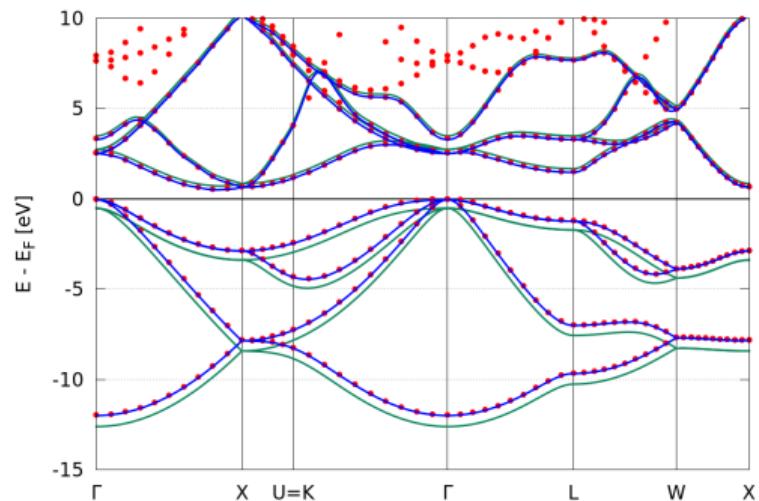
Koopmans functionals: results for solids



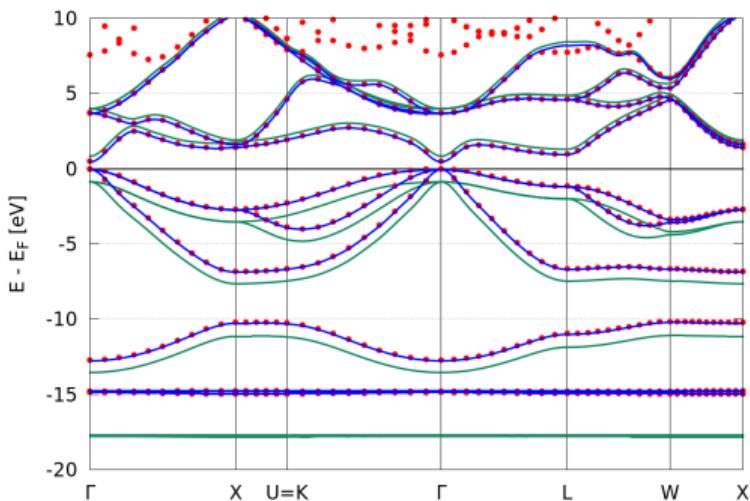
Mean absolute error (eV) across prototypical semiconductors and insulators

	PBE	G ₀ W ₀	KI	KIPZ	QSGW̃
E_{gap}	2.54	0.56	0.27	0.22	0.18
IP	1.09	0.39	0.19	0.21	0.49

Koopmans functionals: results for solids



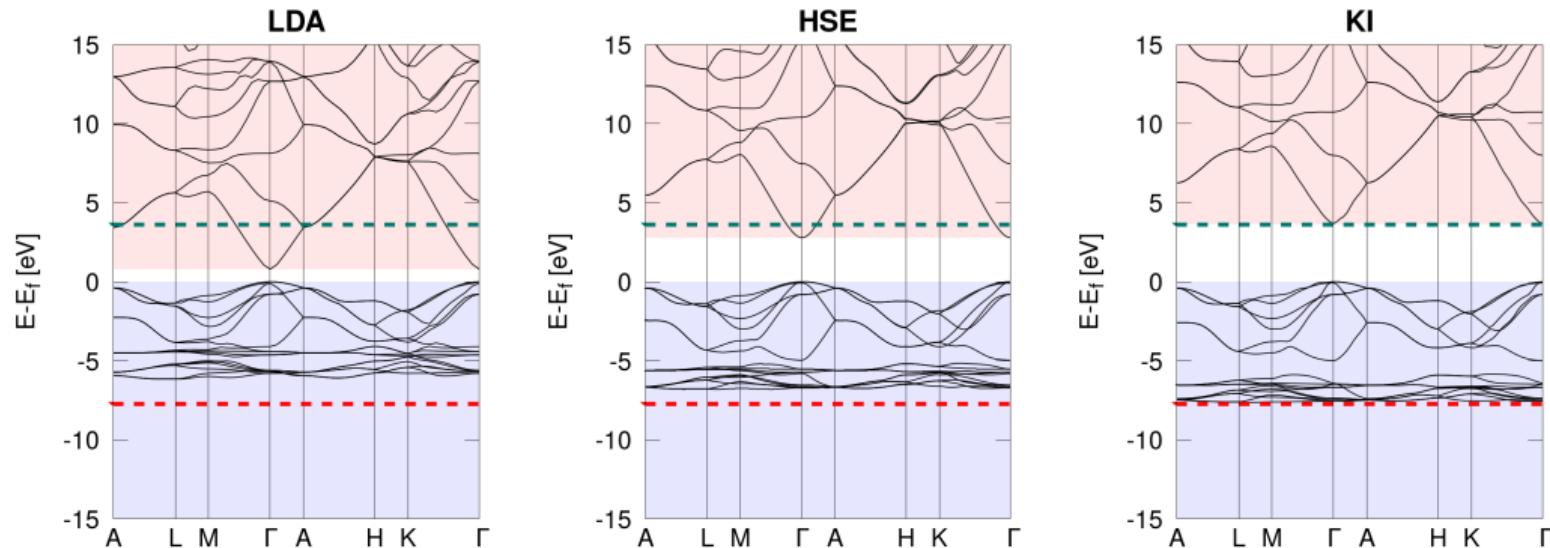
(a) Si, KIPZ



(b) GaAs, KI

		PBE	QSGW	KI	pKIPZ	KIPZ	exp
Si	E_{gap}	0.55	1.24	1.18	1.17	1.19	1.17
GaAs	E_{gap}	0.50	1.61	1.53	1.49	1.50	1.52
	$\langle \varepsilon_d \rangle$	14.9	17.6	16.9	17.7	18.9	

Koopmans functionals: results for solids



ZnO	LDA	HSE	GW_0	$scG\tilde{W}$	KI	exp
E_{gap} (eV)	0.79	2.79	3.0	3.2	3.62	3.60
$\langle \varepsilon_d \rangle$ (eV)	-5.1	-6.1	-6.4	-6.7	-6.9	-7.5/-8.0

`kcw.x` (DFPT implementation) is distributed in Quantum ESPRESSO v7.1 onwards

`kcp.x` (supercell implementation) is available publically

But complex workflows mean that...

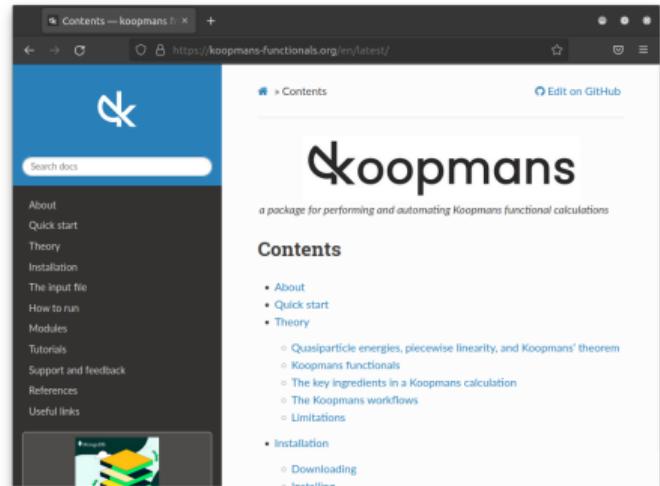
- lots of different codes that need to handshake
- lots of scope for human error
- reproducibility becomes difficult
- expert knowledge required

Our solution...

koopmans

- beta version released earlier this year¹
- implementations of Koopmans functionals
- automated workflows
 - start-to-finish Koopmans calculations
 - Wannierisation
 - dielectric tensor
 - ...
- built on top of ASE²
- under the hood, calls Quantum ESPRESSO
- does not require expert knowledge

koopmans-functionals.org



¹Linscott et al., in prep

²A. H. Larsen et al. *J. Phys. Condens. Matter* 29.27 (2017), 273002

koopmans: the input file

```
{  
    "workflow": {  
        "task": "singlepoint",  
        "functional": "ki",  
        "method": "dscf",  
        "init_orbitals": "mlwfs",  
        "alpha_guess": 0.1  
    },  
    "atoms": {  
        "atomic_positions": {  
            "units": "crystal",  
            "positions": [[{"Si": 0.00, 0.00, 0.00},  
                          {"Si": 0.25, 0.25, 0.25}]]  
        },  
        "cell_parameters": {  
            "periodic": true,  
            "ibrav": 2,  
            "celldm(1)": 10.262  
        }  
    },  
}
```

```
"k_points": {  
    "grid": [8, 8, 8],  
    "path": "LGXKG"  
},  
"calculator_parameters": {  
    "ecutwfc": 60.0,  
    "w90": {  
        "projections": [  
            [{"fsite": [0.125, 0.125, 0.125],  
             "ang_mtm": "sp3"}],  
            [{"fsite": [0.125, 0.125, 0.125],  
             "ang_mtm": "sp3"}]  
        ],  
        "dis_froz_max": 11.5,  
        "dis_win_max": 17.0  
    }  
}
```

koopmans is scriptable

```
from ase.build import bulk
from koopmans.kpoints import Kpoints
from koopmans.projections import ProjectionBlocks
from koopmans.workflows import SinglepointWorkflow

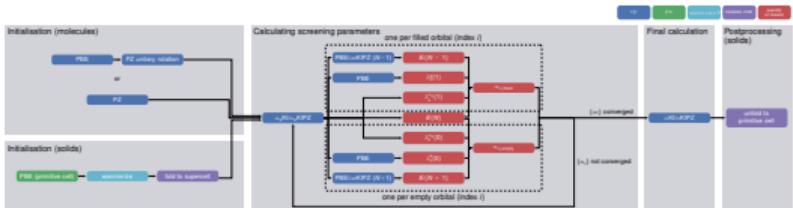
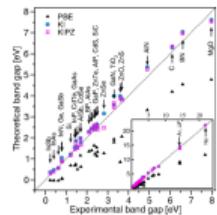
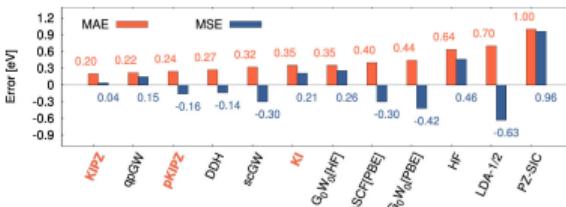
# Use ASE to create bulk silicon
atoms = bulk('Si')

# Define the projections for the Wannierization (same for filled and empty manifold)
si_proj = [{'fsite': [0.25, 0.25, 0.25], 'ang_mtm': 'sp3'}]
si_projs = ProjectionBlocks.from_list([si_proj, si_proj], atoms=atoms)

# Create the workflow
workflow = SinglepointWorkflow(atoms = atoms,
                                projections = si_projs,
                                ecutwfc = 40.0,
                                kpoints = Kpoints(grid=[8, 8, 8], path='LGXKG', cell=atoms.cell),
                                calculator_parameters = {'pw': {'nbnd': 10},
                                                        'w90': {'dis_froz_max': 10.6, 'dis_win_max': 16.9}})

# Run the workflow
workflow.run()
```

Take home messages



- Koopmans functionals are a class of functionals that treat spectral properties on the same footing as total energy differences (via GPWL)
- they can give orbital energies and band structures with comparable accuracy to state-of-the-art GW
- Koopmans functionals are implemented in Quantum ESPRESSO, and we can use koopmans to run these calculations easily

Coming up: a tutorial on Koopmans functionals

But first: Yannick will talk about predicting screening parameters with ML

Follow [Twitter @ed_linscott](#) for updates | Slides available at [GitHub github/elinscott](#)



Nicola Marzari



Nicola Colonna



Riccardo
De Gennaro



Yannick
Schubert



**Swiss National
Science Foundation**



SPARE SLIDES

Recap from earlier

Key idea: construct a functional such that the *variational* orbital energies

$$\varepsilon_i^{\text{Koopmans}} = \langle \varphi_i | H | \varphi_i \rangle = \partial E_{\text{Koopmans}} / \partial f_i$$

are...

- independent of the corresponding occupancies f_i
- equal to the corresponding total energy difference $E_i(N - 1) - E(N)$

zero band gap \rightarrow occupancy matrix for variational orbitals is off-diagonal