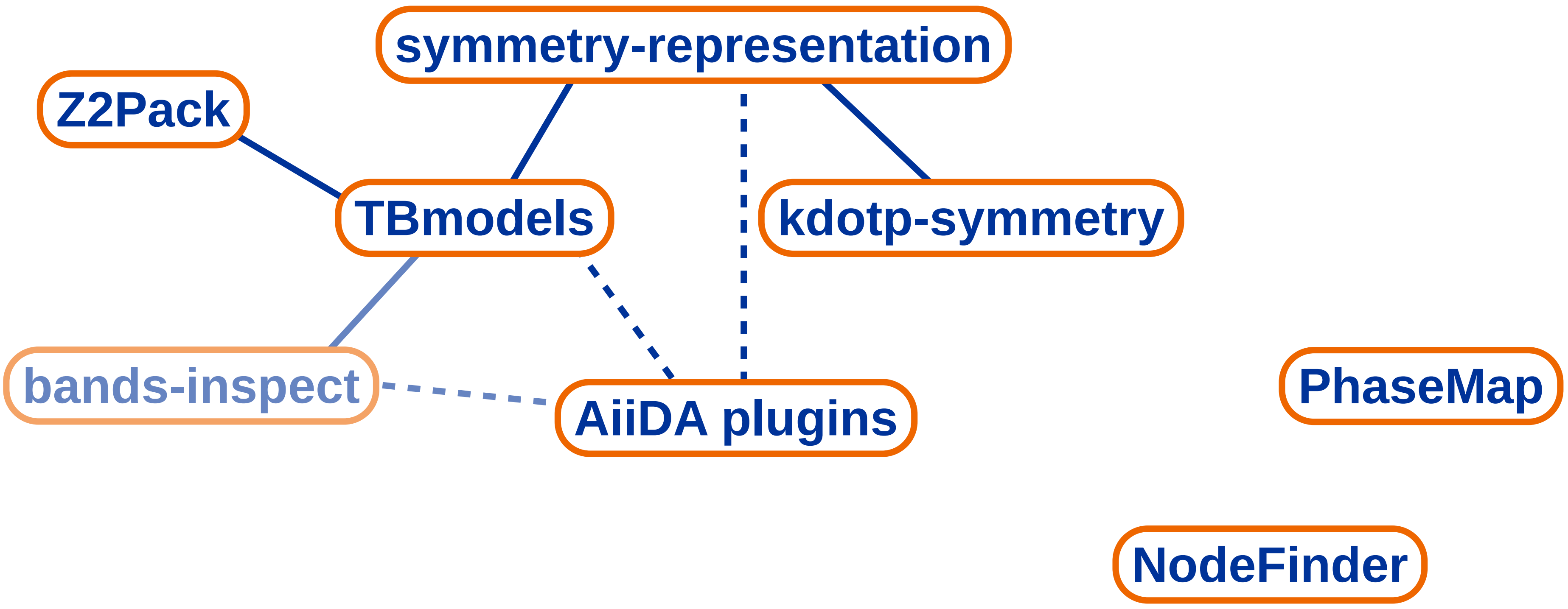
Tools Introduction

Dominik Gresch

31.01.2019

Overview



31.01.2019

symmetry-representation

- **Goal:**

  Describe symmetry operations, create representation matrix

$$\mathcal{H}(\mathbf{k}) = D(g)\mathcal{H}(g^{-1}\mathbf{k})D(g^{-1})$$

needs: representation matrix real-space matrix

TBmodels

- **Goal:**

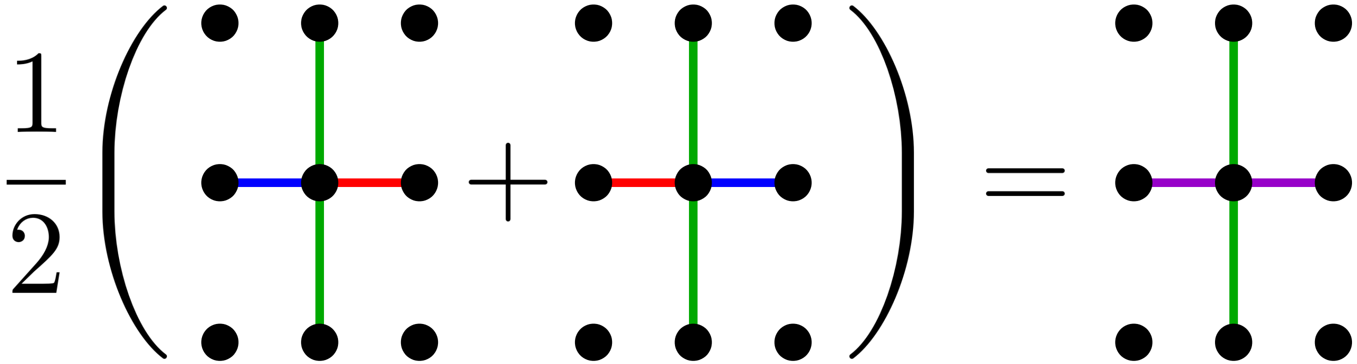
  Evaluate and modify tight-binding models

- **Features:**
  - Efficient evaluation
  - Parse to / from different formats (e.g. Wannier90 output)
  - Symmetrization

TBmodels: Symmetrization

- Calculates group average:

$$\tilde{\mathcal{H}}(\mathbf{k}) = \frac{1}{|G|}\sum_{g\in G} D^{\mathbf{k}}(g)\mathcal{H}(g^{-1}\mathbf{k})D^{\mathbf{k}}(g^{-1})$$



- Condition: Tight-binding *basis* must be complete under symmetry

Gresch et al., Phys. Rev. Materials 2, 103805 (2018)

kdotp-symmetry

- **Goal:**

  Create k·p *basis* which respects given symmetries

$$\mathcal{H}(\mathbf{k}) = D(g)\mathcal{H}(g^{-1}\mathbf{k})D(g^{-1})$$

Z2Pack

- **Goal:**

Calculate topological invariants (Chern number, Z2 invariant)

Tracks Wannier charge center / Wilson loop eigenvalue evolution across a surface

PhaseMap

- **Goal:**

Calculate a phase diagram with minimal number of phase evaluations

Requires a function to evaluate the phase at a given point

NodeFinder

- **Goal:**

  Find and classify all nodal features in a potential landscape (use case: band gap closures)

AiiDA plugins

- What is AiiDA?
  - Workflow / automation framework
- Pro:
  - Allows for high-throughput calculations
  - Creates “provenance graph” of calculations
- Con:
  - Significant learning curve
  - Overhead of creating plugins
  - Additional complexity: non-zero chance of bugs

AiiDA plugins

- Is it for me?
  - For things which could also be done (reasonably) by hand: no
  - Larger / more complex runs: yes

AiiDA plugins

- aiida-tbextraction:
  - plugin for calculating tight-binding models from first-principles calculations
  - optimize Wannier90 energy window
  - includes symmetrization procedure

- Modular workflow design: Link

Documentation: Links

- Z2Pack: http://z2pack.ethz.ch/doc
- Other Tools: http://z2pack.ethz.ch/projects.html
- AiiDA plugins:
  - aiida-optimize.readthedocs.io
  - aiida-strain.readthedocs.io
  - aiida-bands-inspect.readthedocs.io
  - aiida-symmetry-representation.readthedocs.io
  - aiida-tbmodels.readthedocs.io
  - aiida-tbextraction.readthedocs.io

Theory Explanation

- kdotp-symmetry, PhaseMap, NodeFinder: https://doi.org/10.3929/ethz-b-000308602

- Z2Pack: paper, book chapter

- TBmodels symmetrization, AiiDA plugins: https://dx.doi.org/10.1103/PhysRevMaterials.2.103805

Coding Guidelines

- Scientific code is a mess – and that’s fine

- Code to solve problems, *refactor* to create tools

Coding Guidelines

- “Unix philosophy”: small tools which do one thing well
  - In hindsight, maybe fewer separate *repositories* would be simpler

- Design from the user perspective
  - think about how to use the code first
  - write ~~extensive~~ sufficient documentation

Coding Guidelines

- Checkpoints: invaluable for longer-running calculations
  - include them in your design from the start
  - no, `pickle` is not a long-term storage format
- Use unit-testing, static code analyzer, code formatter, …
  - can be a pain to use, but worth it in the long run