This paper describes an approach to automatically evaluate computational thinking skills demonstrated in block-based programming assignments, specifically using Snap!, a visual programming language. The motivation is to provide personalized feedback to students in an scalable manner, as manually reviewing and grading these assignments is labor-intensive.

The authors extract features from the XML representation of students' Snap! programs to capture patterns of abstraction (e.g., functional decomposition) and iteration. This includes measures like repetition of code blocks, nesting/depth of blocks, and number of unique children per parent node. They explore various clustering algorithms (e.g., k-means, Gaussian mixture models) to group submissions with similar features, corresponding to comparable levels of demonstrated computational thinking skills.

The paper presents their methodology, experimental results analyzing the effectiveness of different features and clustering approaches, and discusses limitations and potential future Work. While their current features don't fully capture all aspects of computational thinking, the relative inertia scores suggest promise in using clustering models tailored to individual rubric items. Integrating AI-generated text feedback is proposed as a way to provide more personalized comments.