I. Introduction
Automated program repair (APR) has great potential to reduce bug-fixing effort and many approaches have been proposed in recent years. In this paper, we propose a novel automatic program repair approach, called SimFix, that utilizes two kinds of data sources:
(1) Similar code within faulty program.
(2) Exiting patches from other projects.

II. Motivation
Pros of similar code:
Provide repair guidance with fixing ingredients.
Cons of existing patches:
Cannot cover all repairs under specific context.

III. Approach
Two stage of SimFix:

1. Mining Stage mines a set of frequent abstract modifications from open-source programs.
2. Repairing Stage generates concrete modifications to faulty code via AST differencing with similar code, and take intersection with abstract modifications to rule out invalid modifications.

IV. Evaluation

1. Correct repair
34 bugs were repaired (most number so far).
13 bugs were never repaired by others.

2. Effectiveness of other components.
Existing patches: 12 less correct and 14 more incorrect patches without using existing patches.
Fine-grained code reuse: 17 less bugs would be repaired without fine-grained code reuse (AST node level).