

Motivating Questions

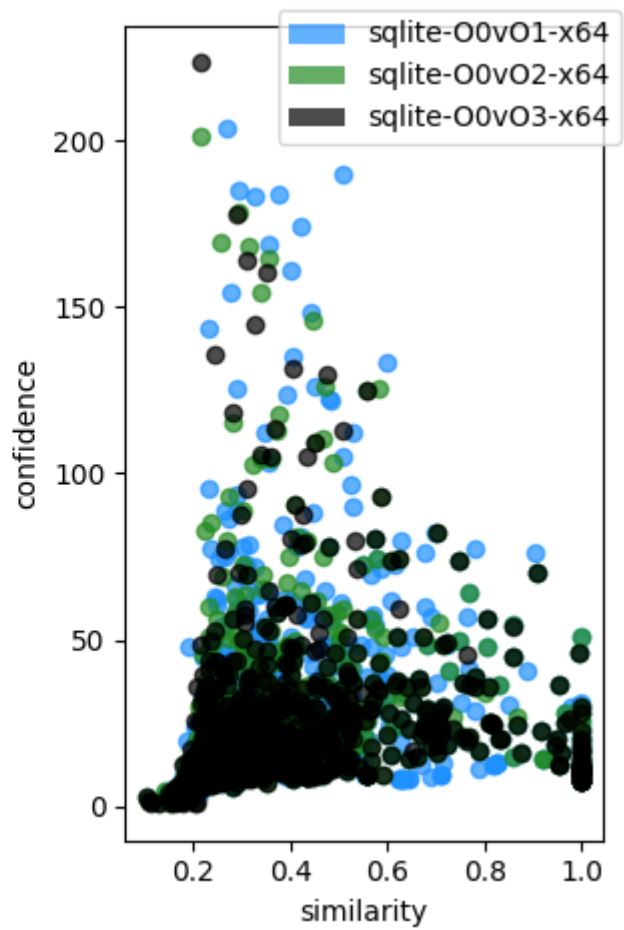
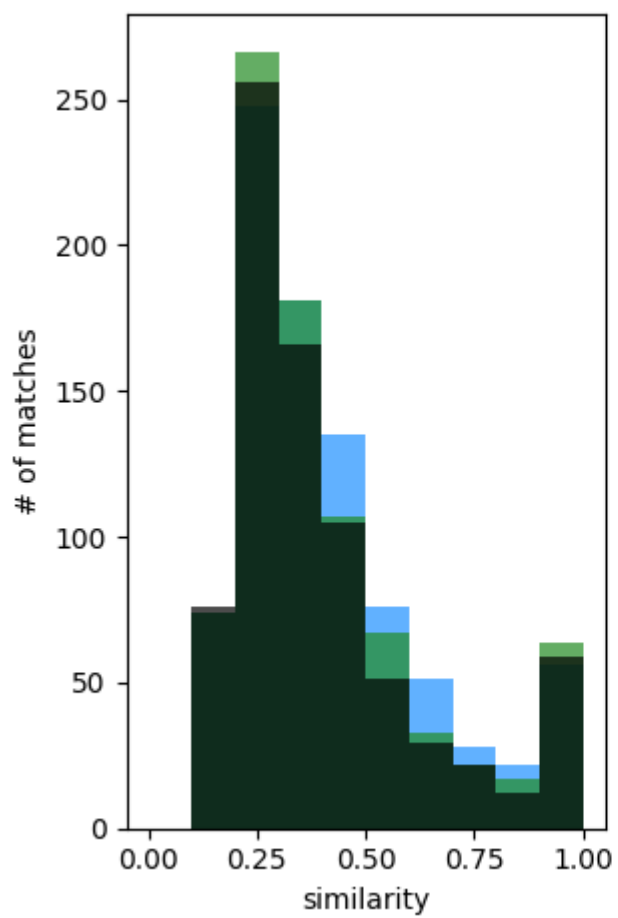
1. Does BSim's performance vary when the *source language varies*? Further, does BSim include *implicit biases based on source-level semantics* of a particular language?
2. How robust are BSim's features to *changes in compilation settings*? Further, do particular compilation settings degrade BSim's performance more than others?
3. Are BSim's weighting and scoring mechanisms sufficient to distinguish false positive matches from true positives? And how many does it produce with a given range?

Experiment 1 - SQLite3 CLI, C, Vary Optimizations

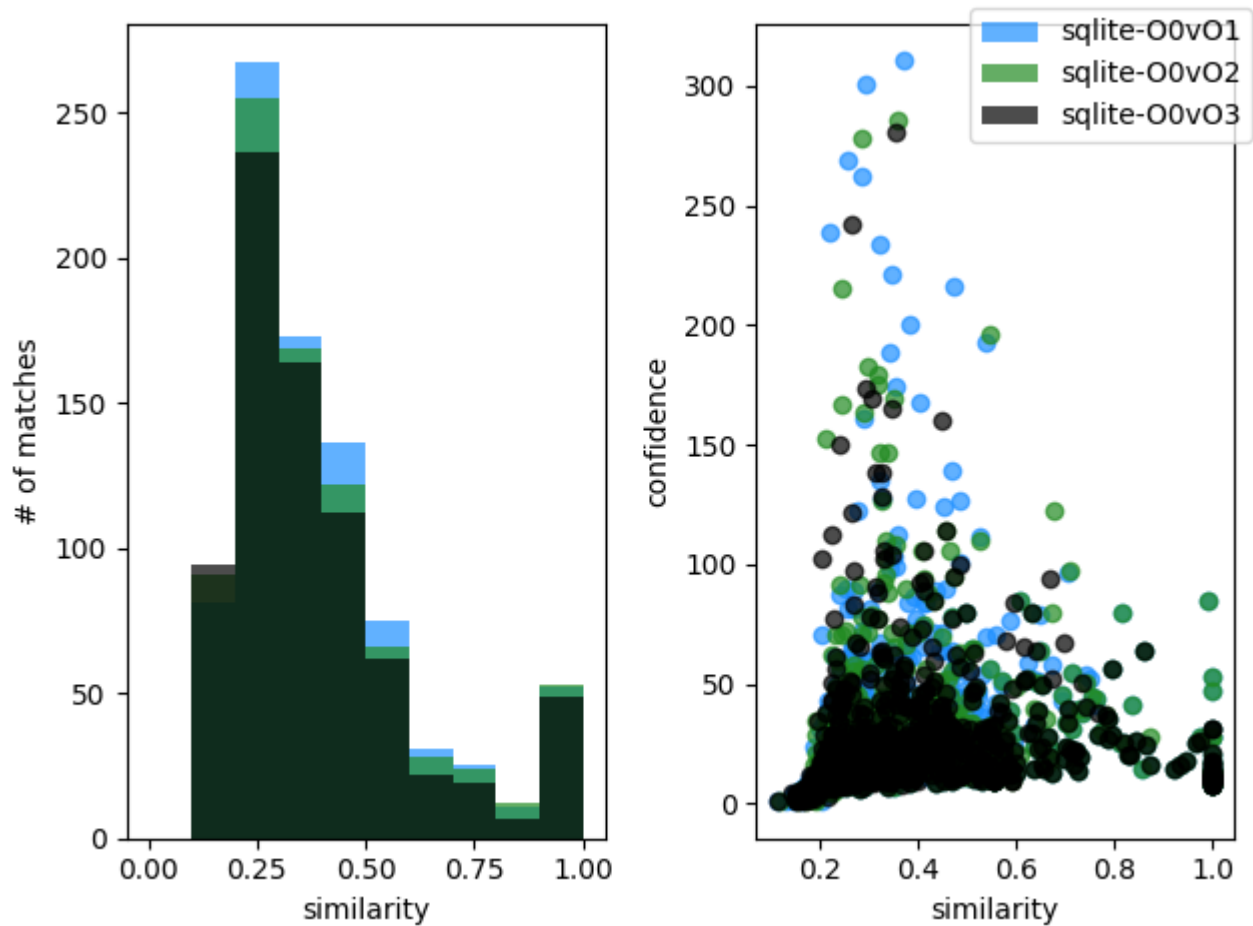
This experiments sets a baseline for BSim's performance on C binaries. We compile the SQLite3 command line tool using Clang at optimization levels O0 - O3. Signatures are generated for O0 and queries are performed on each level against them.

	sqlite-O0vO1-x64	sqlite-O0vO2-x64	sqlite-O0vO3-x64
Avg. Similarity	0.4286	0.4189	0.4121
No. of Functions	1605	1592	1580
No. of Matches	882	843	787
Accuracy	0.5495	0.5295	0.4981

x64 Accuracy



x64 Distribution



ARM Distribution

	sqlite-O0vO1	sqlite-O0vO2	sqlite-O0vO3
Avg. Similarity	0.4112	0.4095	0.3983
No. of Functions	1681	1665	1670
No. of Matches	862	832	775
Accuracy	0.5128	0.4997	0.4641

ARM Accuracy

Experiment 2 - Bat Command Line Tool, Rust, Vary Optimizations

This experiment explores how BSim performs on a modern programming language, Rust. We compile the [Bat](#) command line application with *rustc/cargo* at optimization levels O0-O3. Further, we disable LTO and any symbol stripping. Signatures are again generated for the O0 level and queries are performed at each level against them.

	bat-O0vO0-arm	bat-O0vO1-arm	bat-O0vO2-arm	bat-O0vO3-arm
No. of Functions	497	79	72	71
Top 1 Accuracy	0.706	0.038	0.028	0.028
Top 3 Accuracy	0.755	0.063	0.028	0.042
Top 5 Accuracy	0.777	0.063	0.042	0.042
Top 10 Accuracy	0.793	0.089	0.069	0.07
Top 25 Accuracy	0.849	0.127	0.069	0.07

ARM Accuracy

	bat-O0vO0-x64	bat-O0vO1-x64	bat-O0vO2-x64	bat-O0vO3-x64
No. of Functions	497	71	37	36
Top 1 Accuracy	0.706	0.056	0.027	0.028
Top 3 Accuracy	0.759	0.085	0.027	0.028
Top 5 Accuracy	0.781	0.127	0.054	0.056
Top 10 Accuracy	0.819	0.141	0.108	0.111
Top 25 Accuracy	0.881	0.169	0.108	0.167

x64 Accuracy

Experiment 3 - Hyper Web Server, Rust, Vary Optimizations

To confirm the results presented above, we conducted the same experiment on additional Rust applications. Here, we created a small HTTP server using the [Hyper](#) library, written entirely in Rust. We compile at the varying optimization levels and query only for functions within the Hyper namespace.

	hyper-O0vO0-arm	hyper-O0vO1-arm	hyper-O0vO2-arm	hyper-O0vO3-arm
No. of Functions	274	130	99	99
Top 1 Accuracy	0.668	0.108	0.141	0.131
Top 3 Accuracy	0.759	0.162	0.192	0.172
Top 5 Accuracy	0.796	0.208	0.222	0.202
Top 10 Accuracy	0.828	0.277	0.293	0.273
Top 25 Accuracy	0.869	0.338	0.343	0.323

ARM Accuracy

	hyper-O0vO0-x64	hyper-O0vO1-x64	hyper-O0vO2-x64	hyper-O0vO3-x64
No. of Functions	359	161	121	121
Top 1 Accuracy	0.677	0.161	0.174	0.174
Top 3 Accuracy	0.777	0.224	0.231	0.231
Top 5 Accuracy	0.833	0.248	0.256	0.256
Top 10 Accuracy	0.864	0.28	0.314	0.298
Top 25 Accuracy	0.928	0.323	0.38	0.388

x64 Accuracy

Experiment 4 - Egg Expression Equality Saturation, Rust, Vary optimizations

To further confirmation of the Rust results, we conducted another experiment using the [Egg](#) library, a fully Rust-implemented library for Egraphs based equality saturation. This additional data point covers functionality not previously exercised by the other experiments.

	egg-O0v00	egg-O0v01	egg-O0v02	egg-O0v03
No. of Functions	287	86	72	71
Top 1 Accuracy	0.805	0.081	0.056	0.056
Top 3 Accuracy	0.878	0.14	0.111	0.113
Top 5 Accuracy	0.889	0.163	0.125	0.113
Top 10 Accuracy	0.934	0.174	0.153	0.141
Top 25 Accuracy	0.944	0.244	0.25	0.254

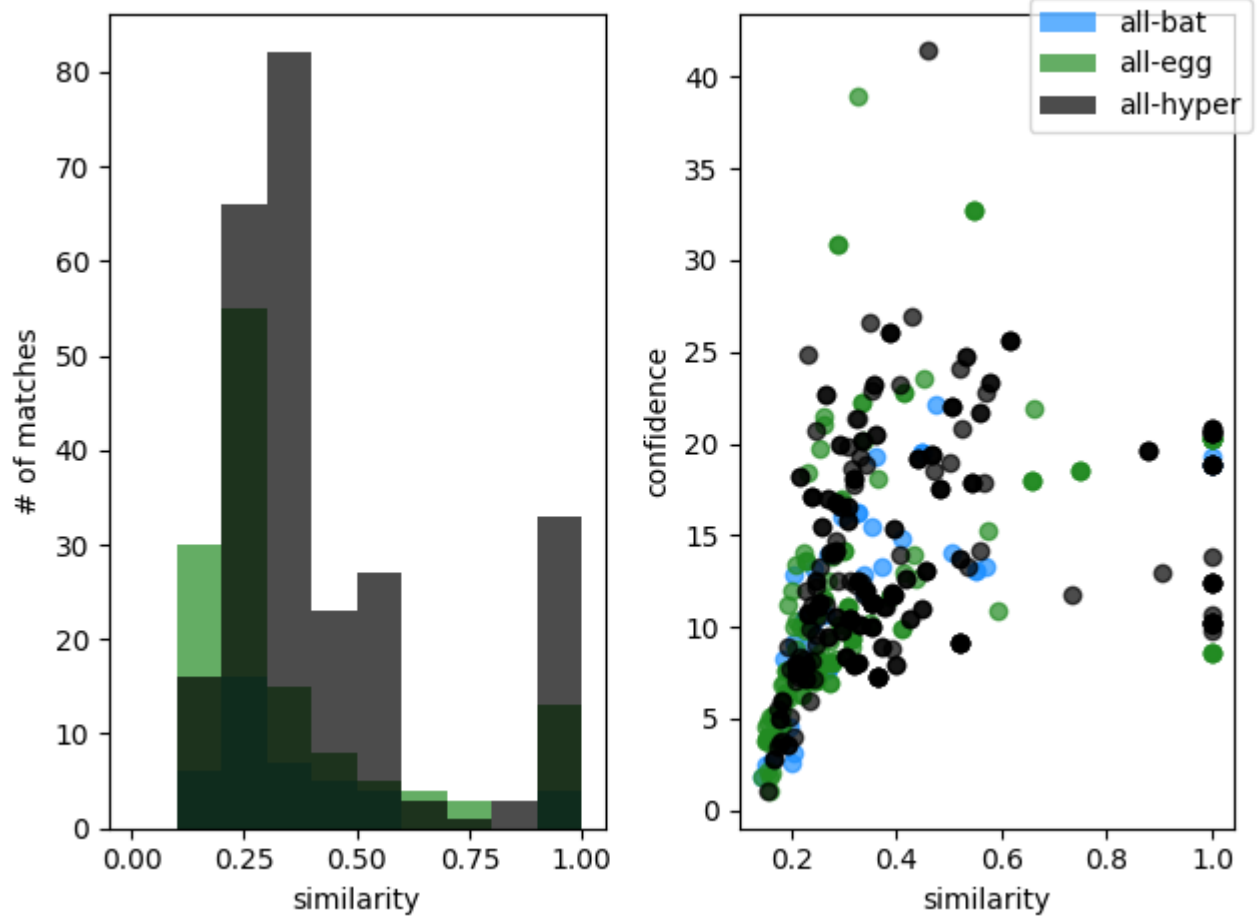
ARM Accuracy

	egg-O0v00	egg-O0v01	egg-O0v02	egg-O0v03
No. of Functions	398	128	113	112
Top 1 Accuracy	0.553	0.102	0.115	0.098
Top 3 Accuracy	0.714	0.148	0.177	0.161
Top 5 Accuracy	0.791	0.156	0.186	0.179
Top 10 Accuracy	0.874	0.188	0.204	0.196
Top 25 Accuracy	0.967	0.234	0.212	0.196

x64 Accuracy

Combining Rust Results

Using all data points from the Rust experiments, generate a similar graph to correlate match accuracy with similarity and confidence



Rust Correlation