An Eclipse-based Integrated and Automated Fault Localization System

Tristan Challener

April 27, 2015

Table of Contents

Motivation

- Debugging is complex and difficult

Motivation

- Debugging is complex and difficult
- Fault localization is the most expensive

Motivation

- Debugging is complex and difficult
- Fault localization is the most expensive
- Current techniques can be improved

Automatic Fault Localization

- Uses per-test coverage analysis

Automatic Fault Localization

- Uses per-test coverage analysis
- Ranks statements by suspiciousness

Automatic Fault Localization

- Uses per-test coverage analysis
- Ranks statements by suspiciousness
- Variety of risk evaluation functions
  Tarantula equation:

$$suspiciousness(e) = 1 - \frac{\frac{failed(e)}{totalfailed}}{\frac{passed(e)}{totalpassed} + \frac{failed(e)}{totalfailed}} \qquad (1)$$

CodeCover Coverage

- Eclipse-compatible coverage analysis tool

CodeCover Coverage

- Eclipse-compatible coverage analysis tool
- Uses existing JUnit test suites

CodeCover Coverage

- Eclipse-compatible coverage analysis tool
- Uses existing JUnit test suites
- Generates coverage information for each test method

CodeCover Coverage

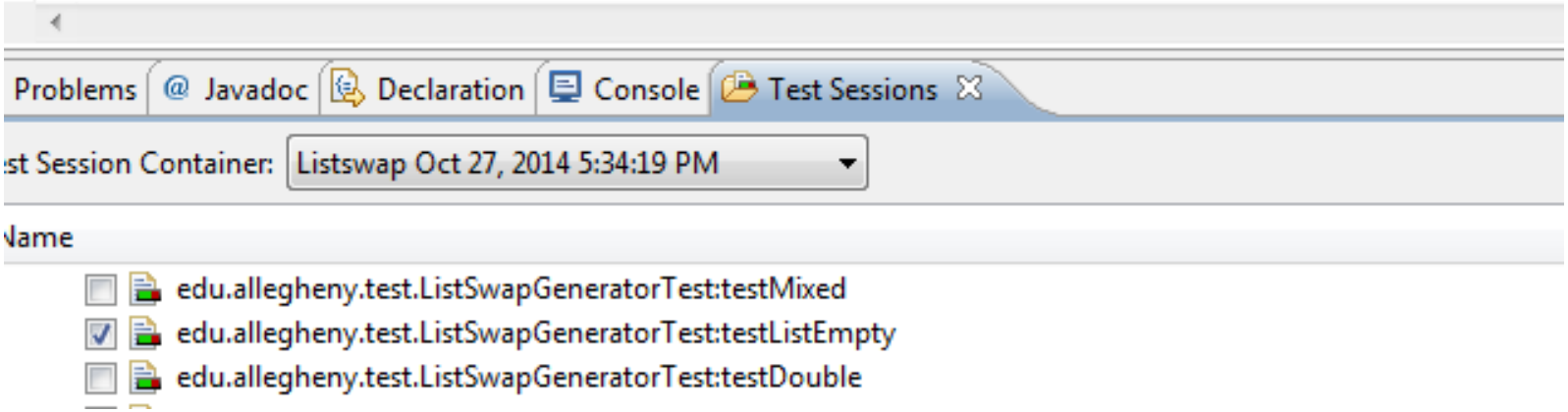
- Eclipse-compatible coverage analysis tool
- Uses existing JUnit test suites
- Generates coverage information for each test method
- Stores output in readable format (XML)

CodeCover

```
public static String listsToString(ArrayList<ArrayList<Object>> list)
{
    String ret = "";
    int listSize = list.size();

    for(int i = 0; i < listSize; i++)
    {
        ret += "L" + (i + 1) + ": \t" + list.get(i);
        if (i < listSize)
            ret += "\n";
    }

    return ret;
}
}
```



MAJOR Mutation

- Introducing faults for experimental evaluation

MAJOR Mutation

- Introducing faults for experimental evaluation
- MAJOR mutation system

MAJOR Mutation

- Introducing faults for experimental evaluation
- MAJOR mutation system
- Representative of real-world faults

Parsing XML

- CodeCover stores information as XML

Parsing XML

- CodeCover stores information as XML
- Parse using Java DOM

Parsing XML

- CodeCover stores information as XML
- Parse using Java DOM
- Contains several pieces of information:

Parsing XML

- CodeCover stores information as XML
- Parse using Java DOM
- Contains several pieces of information:
  - Complete source code
  - Statement definitions
  - List of statements covered by each test method for each file under test

Parsing XML

- CodeCover stores information as XML
- Parse using Java DOM
- Contains several pieces of information:
  - Complete source code
  - Statement definitions
  - List of statements covered by each test method for each file under test

```
<BasicStmnt CovItemId="S4" CovItemPrefix="edu.allegheny.listswap.ListSwapGenerator
<LocList>
<Loc EndOffset="777" SrcFileId="1" StartOffset="735"/>
</LocList>
</BasicStmnt>
```

Coverage

```
<TestCase Comment="" Date="1414365770750"
    Name="edu.allegheny.test.ListSwapGeneratorTest:testString">
<CovList>
<CovPrefix CovItemPrefix="edu.allegheny.listswap.ListSwapGenerator.java">
<Cov CovItemId="B1" Value="5"/>
<Cov CovItemId="B2" Value="1"/>
<Cov CovItemId="L1-2" Value="1"/>
<Cov CovItemId="L2-0" Value="1"/>
<Cov CovItemId="L2-1" Value="1"/>
<Cov CovItemId="L2-2" Value="2"/>
<Cov CovItemId="L3-2" Value="1"/>
<Cov CovItemId="S1" Value="1"/>
<Cov CovItemId="S10" Value="1"/>
<Cov CovItemId="S11" Value="1"/>
<Cov CovItemId="S12" Value="1"/>
<Cov CovItemId="S13" Value="6"/>
<Cov CovItemId="S14" Value="6"/>
<Cov CovItemId="S15" Value="5"/>
<Cov CovItemId="S16" Value="1"/>
<Cov CovItemId="S2" Value="1"/>
<Cov CovItemId="S3" Value="4"/>
```

Intermediate Representation

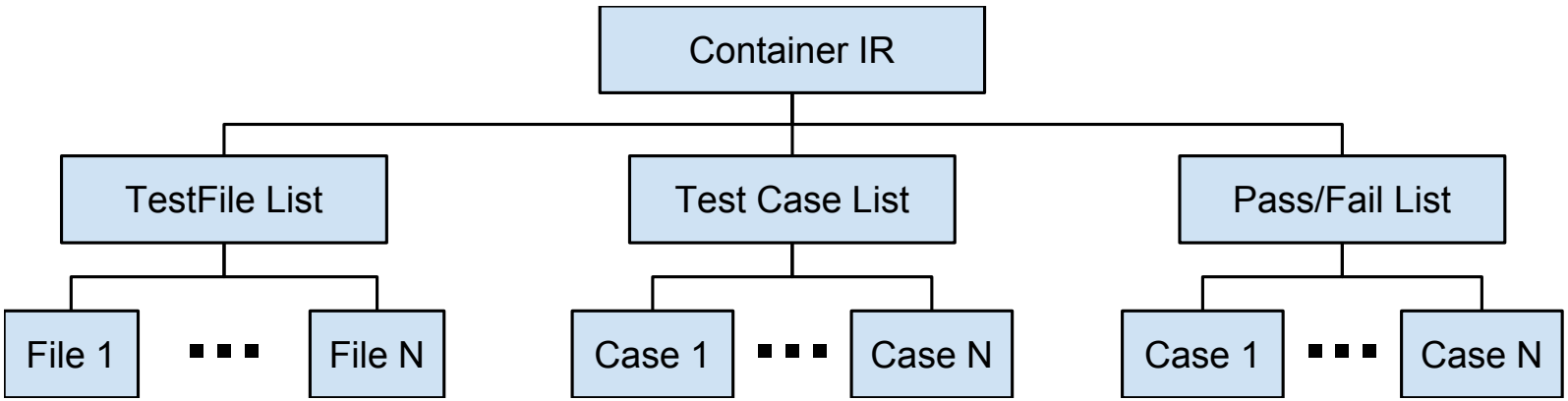
- DOM is overly complex for multi-pass analysis

Intermediate Representation

- DOM is overly complex for multi-pass analysis
- Store information in a more accessible form

Intermediate Representation

- DOM is overly complex for multi-pass analysis
- Store information in a more accessible form



Final Representation

- IR is still not conducive to risk evaluation

Final Representation

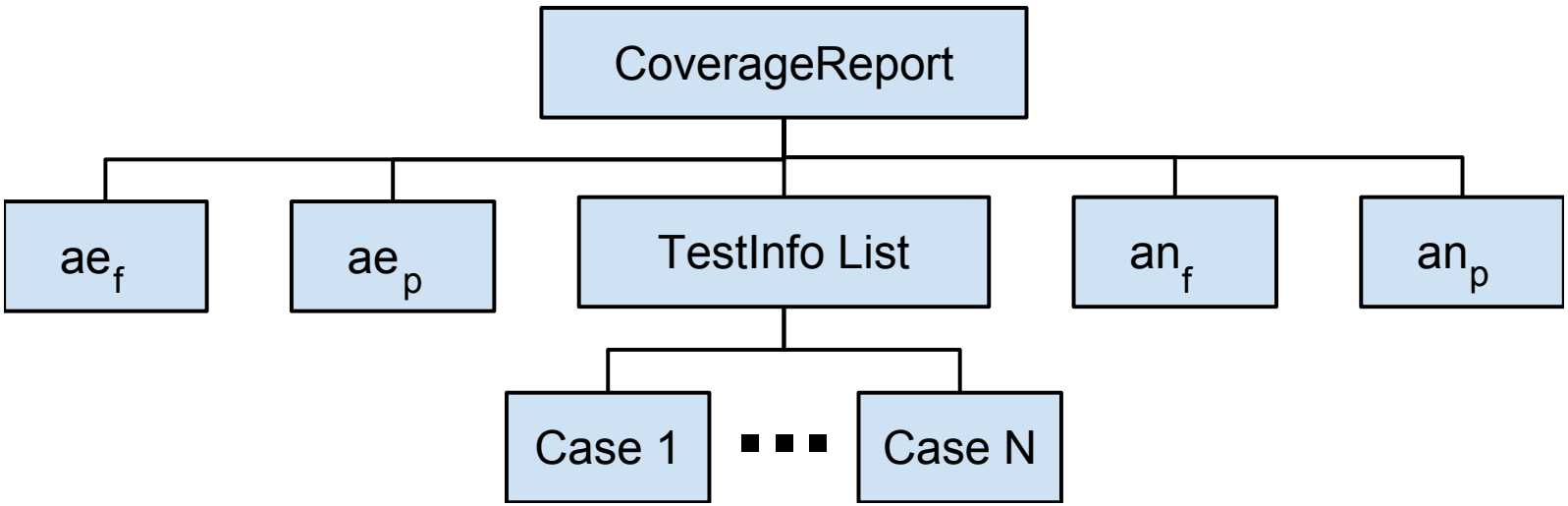
- IR is still not conducive to risk evaluation
- Reformat information into a simpler representation

Final Representation

- IR is still not conducive to risk evaluation
- Reformat information into a simpler representation
- Designed to allow very simple suspiciousness analysis

Final Representation

- IR is still not conducive to risk evaluation
- Reformat information into a simpler representation
- Designed to allow very simple suspiciousness analysis



System Output

- Output data in CSV format

System Output

- Output data in CSV format
- Conform to Tidy Data standard

System Output

- Output data in CSV format
- Conform to Tidy Data standard

```
"Function","StatementID","Filename","Suspiciousness","Rank","StatementCount","CaseApplication","IsFault"
"Jaccard","S2","net.sf.jniinchi.JniInchiAtom.java","1","1","398","jniinchi-119","false"
"Jaccard","S3","net.sf.jniinchi.JniInchiAtom.java","1","2","398","jniinchi-119","false"
"Jaccard","S4","net.sf.jniinchi.JniInchiAtom.java","1","3","398","jniinchi-119","false"
"Jaccard","S5","net.sf.jniinchi.JniInchiAtom.java","1","4","398","jniinchi-119","false"
"Jaccard","S6","net.sf.jniinchi.JniInchiAtom.java","1","5","398","jniinchi-119","false"
"Jaccard","S7","net.sf.jniinchi.JniInchiAtom.java","1","6","398","jniinchi-119","false"
"Jaccard","S8","net.sf.jniinchi.JniInchiAtom.java","1","7","398","jniinchi-119","false"
"Jaccard","S9","net.sf.jniinchi.JniInchiAtom.java","1","8","398","jniinchi-119","false"
"Jaccard","S10","net.sf.jniinchi.JniInchiAtom.java","1","9","398","jniinchi-119","false"
"Jaccard","S11","net.sf.jniinchi.JniInchiAtom.java","1","10","398","jniinchi-119","false"
"Jaccard","S12","net.sf.jniinchi.JniInchiAtom.java","1","11","398","jniinchi-119","true"
```

Case Application

- Obtained case applications from Sarojini Balasubramanian

Case Application

- Obtained case applications from Sarojini Balasubramanian
- Several applications prepared for use with MAJOR

Case Application

- Obtained case applications from Sarojini Balasubramanian
- Several applications prepared for use with MAJOR
- Only one (Jni-InChi) could be processed by CodeCover

Mutant Insertion

- Faults must be introduced for experimentation

Mutant Insertion

- Faults must be introduced for experimentation
- MAJOR analysis information provided by Sarojini

Mutant Insertion

- Faults must be introduced for experimentation
- MAJOR analysis information provided by Sarojini
- Killed mutants of various types selected and inserted into Jni-InChi

```
119:ROR:==(java.lang.Object,java.lang.Object):
  FALSE(java.lang.Object,java.lang.Object):
  net.sf.jniinchi.JniInchiAtom@<init>:117:el == null |==> false
137:STD:<CALL>:<NO-OP>:
  net.sf.jniinchi.JniInchiStructure@addBond:99:
  bondList.add(bond) |==> <NO-OP>
149:LVR:POS:NEG:
  net.sf.jniinchi.JniInchiStereo0D@<init>:79:3 |==> -3
194:COR:||(boolean,boolean):LHS(boolean,boolean):
  net.sf.jniinchi.JniInchiWrapper@checkOptions:183:
  op.startsWith("-") || op.startsWith("/") |==> op.startsWith("-")
197:STD:<CALL>:<NO-OP>:
  net.sf.jniinchi.JniInchiWrapper@checkOptions:189:
  sbOptions.append(flagChar + option.name()) |==> <NO-OP>
```

Results Analysis

- System executed on the CodeCover output for five mutants of Jni-Inchi

Results Analysis

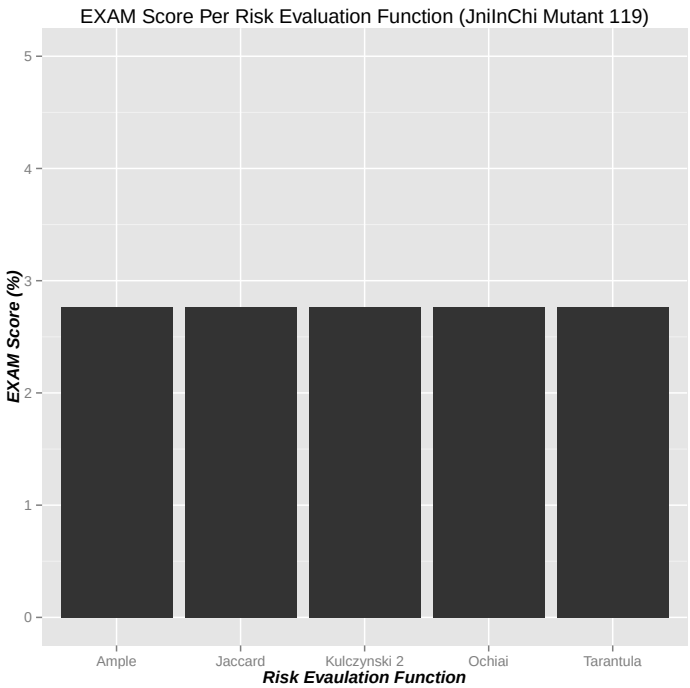
- System executed on the CodeCover output for five mutants of Jni-Inchi
- CSV Output imported into R

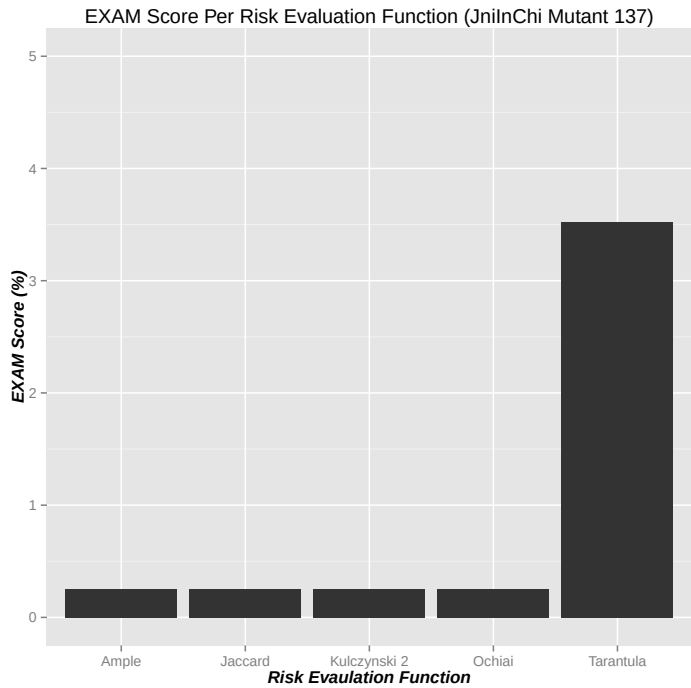
Results Analysis

- System executed on the CodeCover output for five mutants of Jni-Inchi
- CSV Output imported into R
- Plots generated for individual mutants and average overall

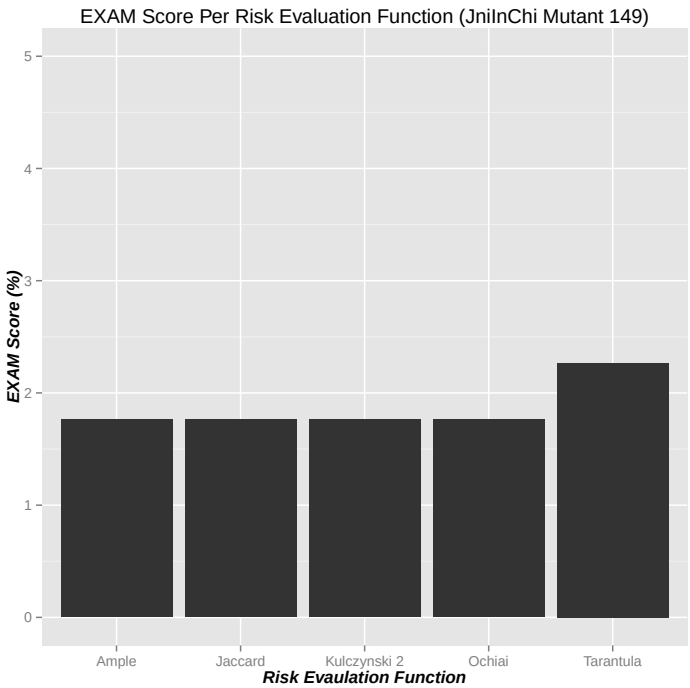
```
library( ggplot2 )
attach( jniinchi_119_fault )
graph_119 <- qplot( Function, Exam, data=jniinchi_119_fault, geom="
   bar",
         ylab="EXAM Score (%)", xlab="Risk Evaulation Function",
       stat="identity", ylim=c( 0,5 ),
       main="EXAM Score Per Risk Evaluation Function (JniInChi
          Mutant 119)",
       sub="JniInChi Mutant 119" )
graph_119 <- graph_119 + theme( axis.title=element_text( face="bold.
   italic" ) )
detach( jniinchi_119_fault )
```

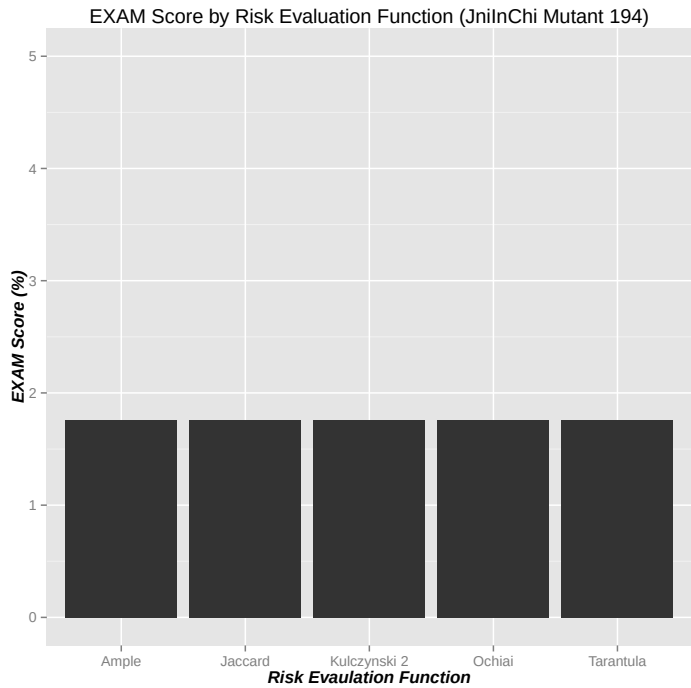
Results


EXAM Score Per Risk Evaluation Function (JniInChi Mutant 119)
EXAM Score (%)
0
1
2
3
4
5
Ample
Jaccard
Kulczynski 2
Ochiai
Tarantula
Risk Evaulation Function

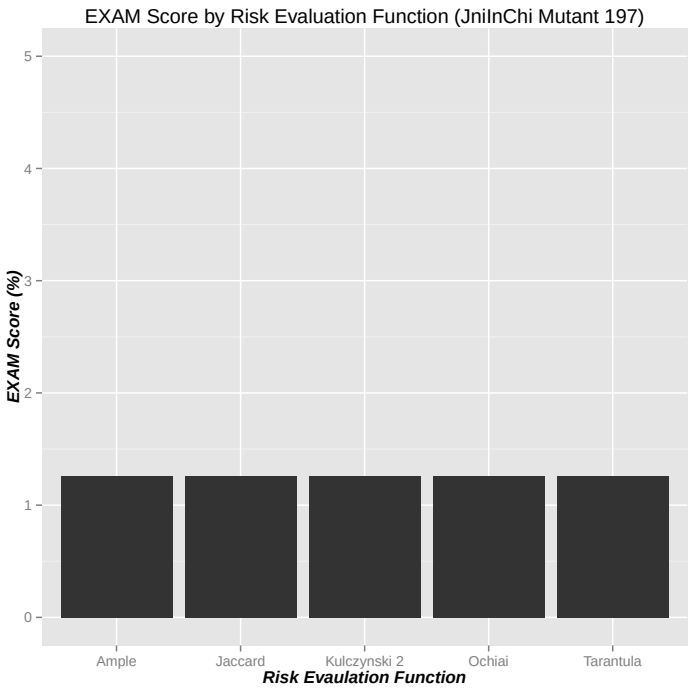

EXAM Score Per Risk Evaluation Function (JniInChi Mutant 137)
EXAM Score (%)
0
1
2
3
4
5
Ample
Jaccard
Kulczynski 2
Ochiai
Tarantula
Risk Evaulation Function

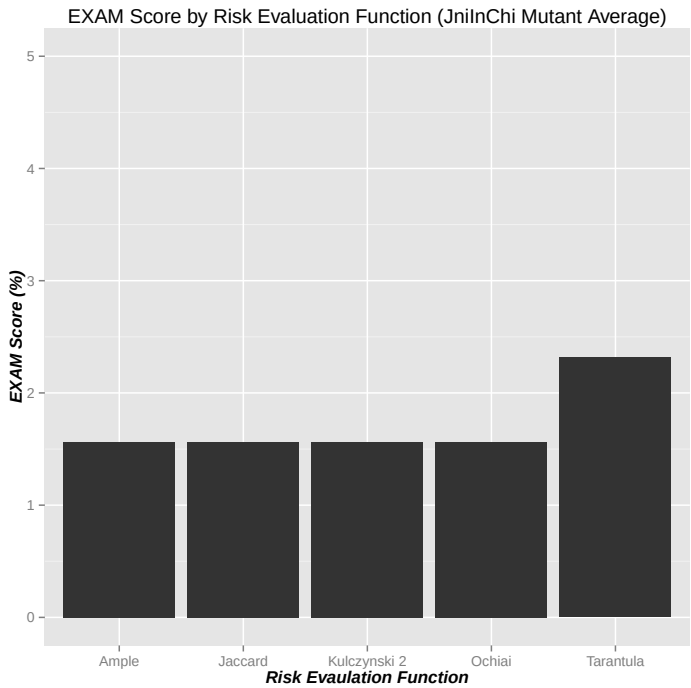
Results


EXAM Score Per Risk Evaluation Function (JniInChi Mutant 149)
EXAM Score (%)
0
1
2
3
4
5
Ample
Jaccard
Kulczynski 2
Ochiai
Tarantula
Risk Evaulation Function


EXAM Score by Risk Evaluation Function (JniInChi Mutant 194)
EXAM Score (%)
0
1
2
3
4
5
Ample
Jaccard
Kulczynski 2
Ochiai
Tarantula
Risk Evaulation Function

Results





Conclusion

- Of our two original goals, the first proved infeasible

Conclusion

- Of our two original goals, the first proved infeasible
- Implemented a system to apply suspiciousness evaluation to coverage data

Conclusion

- Of our two original goals, the first proved infeasible
- Implemented a system to apply suspiciousness evaluation to coverage data
- Completed a small study

Conclusion

- Of our two original goals, the first proved infeasible
- Implemented a system to apply suspiciousness evaluation to coverage data
- Completed a small study
- No statistically significant results generated due to CodeCover limitations

Conclusion

- Of our two original goals, the first proved infeasible
- Implemented a system to apply suspiciousness evaluation to coverage data
- Completed a small study
- No statistically significant results generated due to CodeCover limitations
- Several possible areas for future work identified