**CAmetrics**

1. **发布时间：2018**

文章CAmetrics: A tool for advanced combinatorial analysis and measurement of test sets的发表时间。

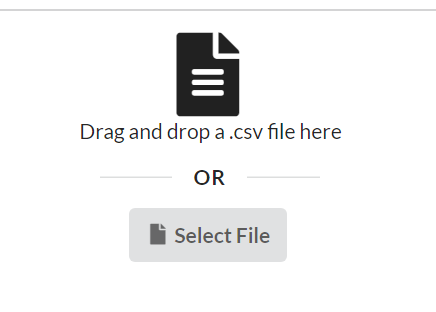
1. **工具使用形式和跨平台特性**

Desktop (CLI) & Web APP

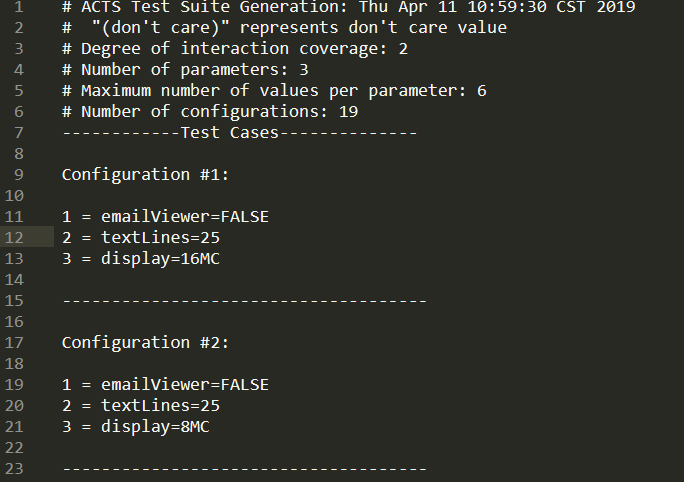
跨平台，命令行提供多平台版本，但是下载windows和MAC版404错误。



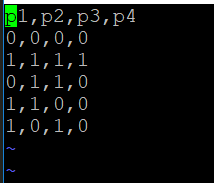
1. **测试生成及其输入输出格式**
   1. **生成算法：**
   2. **支持维度：**
   3. **可变力度：**
   4. **种子：**
   5. **约束：**
   6. **建模语言：**
   7. **输出格式（包括抽象测试用例、自然语言描述测试计划、代码片段等）：**
2. **其它测试活动**
   1. **测试用例优化：**No
   2. **测试执行：**No
   3. **覆盖率评估：**Yes 1-n维覆盖率计算，导入CSV和ACTS格式，web版本仅支持导入CSV

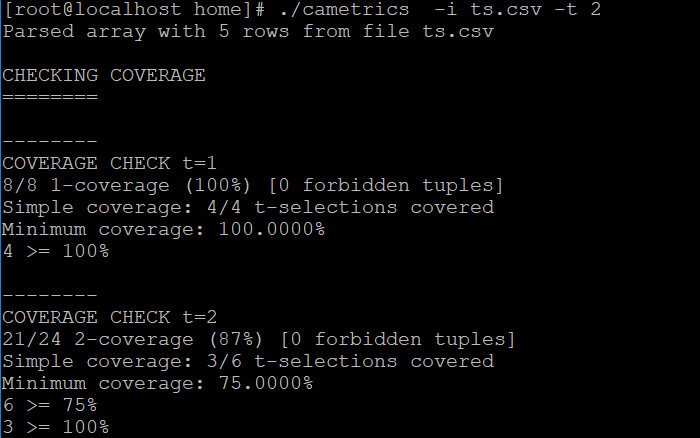


ACTS格式测试用例如下所示

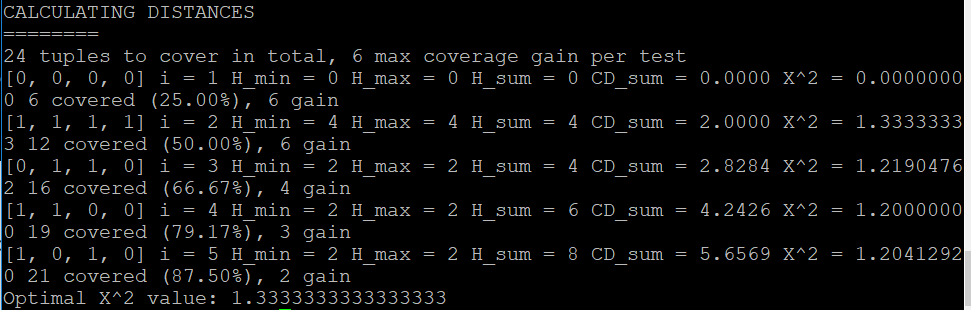


CSV格式如下所示

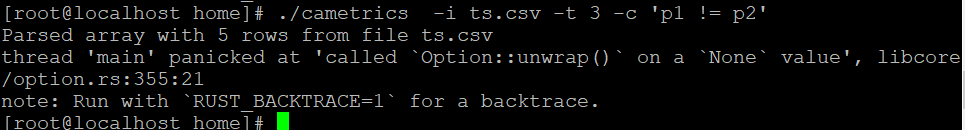




基于距离的评估方法，海明距离，欧式距离，以及α-Balance(α-Balance is defined to be the distribution of t-tuples for each selection of parameters of a test set)

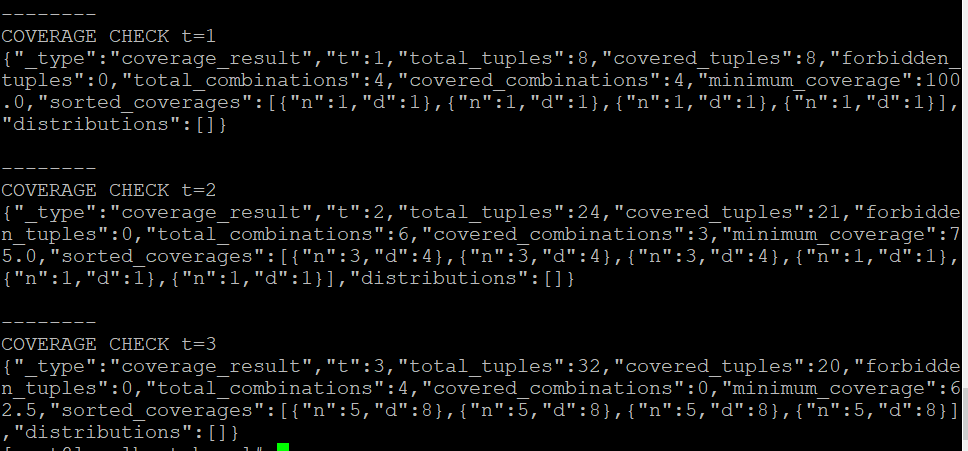


支持约束ABNF语法，见文档最后help说明。当时在使用CSV测试用例时，定义约束后执行失败。其中p1和p2是CSV文件中第一行的参数名。通过在命令中重新定义参数名同样无法尝试约束。



可视化展示覆盖情况，仅web版本支持

支持把输出转化成机器可读的json格式



**故障定位：**No

1. **额外功能**
2. **和其它测试工具的兼容性：**No
3. **工具演化：**

无

附命令行版本help

Usage: ./cametrics

Options:

-h, --help Print help

-t, --strength <strength>

Maximum strength to check for (default: 2)

--no-coverage Skip the computation of coverage. This only works in

case --machine-readable is the chosen output (default:

false)

--dist-strength <strength>

Maximum strength to check value distributions for

(default: do not check value distributions)

-d, --distances Compute per-test distance metrics

-c, --constraint <constraint-spec>

Add a constraint. All constraints must be stated in

the form `[disjunctive-normal-form =>]

conjunctive-normal-form`, e.g. 'P1 != P3' or '(P0 = 2

&& P1 < 3) || P1 = 4 => (P2 > 3 || P3 = 0) && P9 !=

0'.

-p, --parser <acts | csv>

Force a specific parser for your input file. By

default, the 'acts' parser is used unless the input

file name ends in 'csv'.

-s, --separator <sep-char>

Force a specific separator character for the 'csv'

parser. The parser will use ',' as separator by

default.

--machine-readable

Switch to machine-readable output. In this mode,

cametrics output is drastically reduced and all

results are printed as JSON. Note that, depending on

the log level, non-JSON messages will still be

printed. It is thus a reasonable idea to split up the

output into those lines starting with a JSON object

delimiter ({) and all others, using the former to

drive your GUI while displaying the latter to the user

as terminal output.

-l, --log <error | warn | info | debug | trace>

Set the level of output verbosity. The default value

is 'error'. Aside from the values listed here, it

accepts per-module specifications as supported by the

Rust env\_logger library.

-v, --verbose Set the level of output verbosity to 'info'. This is

equivalent to '-l info'.

--param-count <n>

Ignore the parameter names from the input file.

Parameters will be mapped to the names P0, P1, ... Pn.

-m, --param-map <field1,field2,the\_third\_field,lastField>

Ignore the parameter names from the input file and use

a different list instead. The argument to this option

should be a list of parameter names separated by comma

(or the character specified by -s, --separator).

--param-values <param-name>:<v1>,<v2>,...

Specify the values for a single parameter. The format

is '<param-name>:<val1>,<val2>,<val3>...'. This

parameter may be specified multiple times. It is

additive respective to the values found in the input

file (i.e. invalid values are NOT detected) and to

itself (multiple parameters for the same <param-name>

are valid).

--param-values-count <param-name>:<n>

A shortcut to add parameter values to a single

parameter. The format is '<param-name>:<n>', which

adds the parameter values 0,1,..,n to the given

parameter. All other notes for --param-values apply

here.

-i, --input <FILE> Path to the file containing the array to be checked.

If you do not use this parameter, the first free

argument will be used. If you specify '-' as the input

file, cametrics will read from stdin.

--prefix <PATH> Prefix path for input files. Used internally for

compatibility with Docker.

--pwd <PATH> Internal use only. Used by the Docker wrapper to pass

the current working directory to support relative

paths.

--fast Enable fast (but memory-intensive) computation of

coverage measurements. This mode processes tests one

line at a time, storing occurrences of tuples in an

indicator vector with v^t \* (k choose t) elements.

This is the default.

--light Decrease memory usage for coverage measurements, at

the cost of slower computation. This mode only stores

counters for covered and total tuples for all (k

choose t) parameter combinations, but must traverse

the test set more often. Mutually exclusive with

--fast.

--bool Like --light, but do not keep per-combination counters

in memory, further reducing memory usage (but

providing less meaningful output)

--dont-care Specify a specific string as a don't-care value.

Typically such values are denoted as '\*' or '-'.

Default is no don't-care value.

该工具WEB版本相比于命令行精简了如下功能：

* Per-test distance metrics including Hamming, Cartesian and Modified Chi-Square Distance
* Tuple Distribution (i.e. how often each t-tuple appears in the test set), also called Alpha Balance
* Support for more exotic input formats and variations (e.g. TSV for tab-separated input files) as well as custom parameter names
* Multithreading and full access to available memory