

## Markov Model of Keyboard Tokens

0

Generated by Doxygen 1.8.11



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Packages . . . . .	1
<b>2</b>	<b>Hierarchical Index</b>	<b>3</b>
2.1	Class Hierarchy . . . . .	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List . . . . .	5
<b>4</b>	<b>File Index</b>	<b>7</b>
4.1	File List . . . . .	7
<b>5</b>	<b>Namespace Documentation</b>	<b>9</b>
5.1	Package components . . . . .	9
5.2	Package data_analysis . . . . .	9
5.3	Package gui . . . . .	9
5.4	Package junit . . . . .	10
5.5	Package rank . . . . .	10
5.6	Package runtime . . . . .	10
5.7	Package test . . . . .	10
5.8	Package trie . . . . .	10

<b>6</b>	<b>Class Documentation</b>	<b>11</b>
6.1	components.Chain Class Reference	11
6.1.1	Detailed Description	12
6.1.2	Constructor & Destructor Documentation	12
6.1.2.1	Chain(int window, int token, int threshold, int model_size)	12
6.1.2.2	Chain(Chain c)	12
6.1.3	Member Function Documentation	12
6.1.3.1	add_touch(Touch touch)	12
6.1.3.2	add_touch_list(List< Touch > t)	12
6.1.3.3	compare_to(Chain auth_chain)	12
6.1.3.4	compute_uncomputed()	12
6.1.3.5	get_distribution()	12
6.1.3.6	get_key_distribution()	12
6.1.3.7	get_model_size()	13
6.1.3.8	get_threshold()	13
6.1.3.9	get_token()	13
6.1.3.10	get_tokens()	13
6.1.3.11	get_touch_probability(Window w, Touch t)	13
6.1.3.12	get_touches()	13
6.1.3.13	get_window()	13
6.1.3.14	get_windows()	13
6.1.3.15	is_touch_in_key_distribution(Touch touch)	13
6.1.3.16	output_to_csv(String file_name)	13
6.1.3.17	reset()	13
6.1.3.18	set_distribution(Distribution distribution, List< Distribution > key_distribution)	13
6.1.3.19	toString()	14
6.2	runtime.ChainBuilder Class Reference	14
6.2.1	Detailed Description	14
6.2.2	Constructor & Destructor Documentation	14
6.2.2.1	ChainBuilder()	14

6.2.2.2	<a href="#">ChainBuilder(int window, int token, int threshold, int user_model_size, int auth_model_size)</a>	14
6.2.3	<a href="#">Member Function Documentation</a>	15
6.2.3.1	<a href="#">authenticate()</a>	15
6.2.3.2	<a href="#">build_chain_from_csv(File file)</a>	15
6.2.3.3	<a href="#">get_authenticate_state()</a>	15
6.2.3.4	<a href="#">get_authenticate_thread()</a>	15
6.2.3.5	<a href="#">handle_touch(Touch touch)</a>	15
6.2.3.6	<a href="#">parse_csv(File file)</a>	15
6.3	<a href="#">runtime.CompareChains Class Reference</a>	15
6.3.1	<a href="#">Detailed Description</a>	16
6.3.2	<a href="#">Constructor &amp; Destructor Documentation</a>	16
6.3.2.1	<a href="#">CompareChains(Chain user_chain, Chain auth_chain)</a>	16
6.3.3	<a href="#">Member Function Documentation</a>	16
6.3.3.1	<a href="#">get_auth_complete()</a>	16
6.3.3.2	<a href="#">get_auth_probability()</a>	16
6.3.3.3	<a href="#">get_auth_result()</a>	16
6.3.3.4	<a href="#">run()</a>	17
6.3.4	<a href="#">Member Data Documentation</a>	17
6.3.4.1	<a href="#">auth_chain</a>	17
6.3.4.2	<a href="#">authentication_probability</a>	17
6.3.4.3	<a href="#">complete</a>	17
6.3.4.4	<a href="#">is_authentic</a>	17
6.3.4.5	<a href="#">user_chain</a>	17
6.4	<a href="#">rank.CompareChainsRank Class Reference</a>	17
6.4.1	<a href="#">Constructor &amp; Destructor Documentation</a>	18
6.4.1.1	<a href="#">CompareChainsRank(Chain user_chain, Chain auth_chain)</a>	18
6.4.2	<a href="#">Member Function Documentation</a>	18
6.4.2.1	<a href="#">run()</a>	18
6.5	<a href="#">runtime.ChainBuilder.CompareMethod Enum Reference</a>	18
6.5.1	<a href="#">Member Data Documentation</a>	18

6.5.1.1	PROBABILITY_VECTOR_DIFFERENCE . . . . .	18
6.6	rank.CompleteProbability Class Reference . . . . .	18
6.6.1	Detailed Description . . . . .	18
6.6.2	Constructor & Destructor Documentation . . . . .	19
6.6.2.1	CompleteProbability(Chain chain) . . . . .	19
6.6.3	Member Function Documentation . . . . .	19
6.6.3.1	compute_probability() . . . . .	19
6.7	runtime.Operation_thread.Computation Enum Reference . . . . .	19
6.7.1	Member Data Documentation . . . . .	19
6.7.1.1	DISTRIBUTION . . . . .	19
6.7.1.2	KEY_DISTRIBUTION . . . . .	19
6.7.1.3	PROBABILITY . . . . .	19
6.7.1.4	TOKEN . . . . .	19
6.7.1.5	WINDOW . . . . .	19
6.8	test.Main.TestFiles.Concentration Enum Reference . . . . .	20
6.8.1	Constructor & Destructor Documentation . . . . .	20
6.8.1.1	Concentration(String description, int identifier, double value) . . . . .	20
6.8.2	Member Function Documentation . . . . .	20
6.8.2.1	get_identifier() . . . . .	20
6.8.2.2	get_value() . . . . .	20
6.8.2.3	toString() . . . . .	20
6.8.3	Member Data Documentation . . . . .	20
6.8.3.1	HIGH . . . . .	20
6.8.3.2	LOW . . . . .	20
6.8.3.3	MEDIUM . . . . .	20
6.9	test.Main.TestFiles.Distribution Enum Reference . . . . .	21
6.9.1	Constructor & Destructor Documentation . . . . .	21
6.9.1.1	Distribution(String description, int identifier, double value) . . . . .	21
6.9.2	Member Function Documentation . . . . .	21
6.9.2.1	get_identifier() . . . . .	21

6.9.2.2	<a href="#">get_value()</a> . . . . .	21
6.9.2.3	<a href="#">toString()</a> . . . . .	21
6.9.3	<a href="#">Member Data Documentation</a> . . . . .	21
6.9.3.1	<a href="#">ABNORMAL</a> . . . . .	21
6.9.3.2	<a href="#">NORMAL</a> . . . . .	21
6.9.3.3	<a href="#">RANDOM</a> . . . . .	21
6.10	<a href="#">components.Distribution Class Reference</a> . . . . .	22
6.10.1	<a href="#">Detailed Description</a> . . . . .	22
6.10.2	<a href="#">Constructor &amp; Destructor Documentation</a> . . . . .	22
6.10.2.1	<a href="#">Distribution(List&lt; Touch &gt; touches)</a> . . . . .	22
6.10.2.2	<a href="#">Distribution(List&lt; Touch &gt; touches, int keycode)</a> . . . . .	22
6.10.2.3	<a href="#">Distribution(Distribution d)</a> . . . . .	22
6.10.3	<a href="#">Member Function Documentation</a> . . . . .	22
6.10.3.1	<a href="#">equals(Object o)</a> . . . . .	22
6.10.3.2	<a href="#">get_average()</a> . . . . .	23
6.10.3.3	<a href="#">get_keycode()</a> . . . . .	23
6.10.3.4	<a href="#">get_max()</a> . . . . .	23
6.10.3.5	<a href="#">get_min()</a> . . . . .	23
6.10.3.6	<a href="#">get_standard_deviation()</a> . . . . .	23
6.10.3.7	<a href="#">update(List&lt; Touch &gt; touches)</a> . . . . .	23
6.11	<a href="#">test.Main Class Reference</a> . . . . .	23
6.11.1	<a href="#">Detailed Description</a> . . . . .	23
6.11.2	<a href="#">Member Function Documentation</a> . . . . .	23
6.11.2.1	<a href="#">main(String args[])</a> . . . . .	23
6.12	<a href="#">gui.Marcov_console_panel Class Reference</a> . . . . .	24
6.12.1	<a href="#">Constructor &amp; Destructor Documentation</a> . . . . .	24
6.12.1.1	<a href="#">Marcov_console_panel()</a> . . . . .	24
6.13	<a href="#">gui.Marcov_file_display_panel Class Reference</a> . . . . .	24
6.13.1	<a href="#">Constructor &amp; Destructor Documentation</a> . . . . .	24
6.13.1.1	<a href="#">Marcov_file_display_panel()</a> . . . . .	24

6.14	<a href="#">gui.Marcov_frame Class Reference</a>	25
6.14.1	<a href="#">Constructor &amp; Destructor Documentation</a>	25
6.14.1.1	<a href="#">Marcov_frame()</a>	25
6.14.2	<a href="#">Member Function Documentation</a>	25
6.14.2.1	<a href="#">close()</a>	25
6.15	<a href="#">gui.Marcov_options_panel Class Reference</a>	25
6.15.1	<a href="#">Constructor &amp; Destructor Documentation</a>	26
6.15.1.1	<a href="#">Marcov_options_panel()</a>	26
6.16	<a href="#">data_analysis.Model_compare Class Reference</a>	26
6.16.1	<a href="#">Detailed Description</a>	26
6.16.2	<a href="#">Member Function Documentation</a>	26
6.16.2.1	<a href="#">main(String[] args)</a>	26
6.17	<a href="#">data_analysis.Model_compare_thread Class Reference</a>	26
6.17.1	<a href="#">Detailed Description</a>	27
6.17.2	<a href="#">Constructor &amp; Destructor Documentation</a>	27
6.17.2.1	<a href="#">Model_compare_thread(String base_data_path, String auth_data_path, int base_model_size, int auth_model_size, int window_size, int token_size, int threshold)</a>	27
6.17.3	<a href="#">Member Function Documentation</a>	27
6.17.3.1	<a href="#">get_auth_data_path()</a>	27
6.17.3.2	<a href="#">get_auth_model_size()</a>	27
6.17.3.3	<a href="#">get_auth_probability_list()</a>	27
6.17.3.4	<a href="#">get_base_data_path()</a>	27
6.17.3.5	<a href="#">get_base_model_size()</a>	27
6.17.3.6	<a href="#">get_threshold()</a>	27
6.17.3.7	<a href="#">get_token_size()</a>	28
6.17.3.8	<a href="#">get_window_size()</a>	28
6.17.3.9	<a href="#">run()</a>	28
6.17.4	<a href="#">Member Data Documentation</a>	28
6.17.4.1	<a href="#">average_authentication_probability</a>	28
6.17.4.2	<a href="#">max_authentication_probability</a>	28



6.17.4.3	<a href="#">min_authentication_probability</a>	28
6.18	<a href="#">runtime.Operation_thread Class Reference</a>	28
6.18.1	<a href="#">Constructor &amp; Destructor Documentation</a>	28
6.18.1.1	<a href="#">Operation_thread(Chain chain, Computation computation)</a>	28
6.18.2	<a href="#">Member Function Documentation</a>	28
6.18.2.1	<a href="#">run()</a>	28
6.19	<a href="#">test.Main.TestFiles.PressureAmount Enum Reference</a>	29
6.19.1	<a href="#">Constructor &amp; Destructor Documentation</a>	29
6.19.1.1	<a href="#">PressureAmount(String description, int identifier, double value)</a>	29
6.19.2	<a href="#">Member Function Documentation</a>	29
6.19.2.1	<a href="#">get_identifier()</a>	29
6.19.2.2	<a href="#">get_value()</a>	29
6.19.2.3	<a href="#">toString()</a>	29
6.19.3	<a href="#">Member Data Documentation</a>	29
6.19.3.1	<a href="#">HIGH</a>	29
6.19.3.2	<a href="#">LOW</a>	29
6.19.3.3	<a href="#">MEDIUM</a>	29
6.20	<a href="#">test.Print_model Class Reference</a>	30
6.20.1	<a href="#">Member Function Documentation</a>	30
6.20.1.1	<a href="#">main(String[] args)</a>	30
6.21	<a href="#">gui.StartGUI Class Reference</a>	30
6.21.1	<a href="#">Member Function Documentation</a>	30
6.21.1.1	<a href="#">exit()</a>	30
6.21.1.2	<a href="#">main(String[] args)</a>	30
6.22	<a href="#">runtime.ChainBuilder.State Enum Reference</a>	30
6.22.1	<a href="#">Member Data Documentation</a>	31
6.22.1.1	<a href="#">IN_PROGRESS</a>	31
6.22.1.2	<a href="#">SUCCESS</a>	31
6.23	<a href="#">data_analysis.Statistics Class Reference</a>	31
6.23.1	<a href="#">Member Function Documentation</a>	31

6.23.1.1	<code>authentication_accuracy(double authentication_percentage, List&lt; Double &gt; should_authenticate_percentages, List&lt; Double &gt; should_not_authenticate_percentages)</code>	31
6.23.1.2	<code>best_authentication_percentage(List&lt; Double &gt; should_authenticate_percentages, List&lt; Double &gt; should_not_authenticate_percentages)</code>	31
6.23.1.3	<code>equal_false_positive_negative_authentication_percentage(List&lt; Double &gt; should_authenticate_percentages, List&lt; Double &gt; should_not_authenticate_percentages)</code>	31
6.23.1.4	<code>false_negative_percentage(double authentication_percentage, List&lt; Double &gt; should_authenticate_percentages, List&lt; Double &gt; should_not_authenticate_percentages)</code>	31
6.23.1.5	<code>false_positive_percentage(double authentication_percentage, List&lt; Double &gt; should_authenticate_percentages, List&lt; Double &gt; should_not_authenticate_percentages)</code>	31
6.23.1.6	<code>main(String args[])</code>	31
6.23.1.7	<code>minimize_false_positive_authentication_percentage(List&lt; Double &gt; should_authenticate_percentages, List&lt; Double &gt; should_not_authenticate_percentages)</code>	31
6.24	<code>components.Token Class Reference</code>	32
6.24.1	Detailed Description	32
6.24.2	Constructor & Destructor Documentation	32
6.24.2.1	<code>Token(Distribution distribution, int total_tokens, int token_index, double standard_deviations, Type type)</code>	32
6.24.2.2	<code>Token(Distribution distribution, int total_tokens, int token_index, Type type)</code>	33
6.24.2.3	<code>Token(double range_min, double range_max, int total_tokens, int token_index, Type type)</code>	33
6.24.3	Member Function Documentation	33
6.24.3.1	<code>contains(Touch touch)</code>	33
6.24.3.2	<code>equals(Object o_t)</code>	33
6.24.3.3	<code>get_acceptable_wildcards(int total_items)</code>	33
6.24.3.4	<code>get_max()</code>	33
6.24.3.5	<code>get_min()</code>	33
6.24.3.6	<code>get_total_wildcards()</code>	34
6.24.3.7	<code>increment_high_wildcards()</code>	34
6.24.3.8	<code>increment_low_wildcards()</code>	34
6.24.3.9	<code>is_high_wildcard(Touch touch)</code>	34
6.24.3.10	<code>is_low_wildcard(Touch touch)</code>	34

6.25	components.Touch Class Reference	34
6.25.1	Detailed Description	35
6.25.2	Constructor & Destructor Documentation	35
6.25.2.1	Touch(int keycode, double pressure, long timestamp)	35
6.25.2.2	Touch(Touch t)	35
6.25.3	Member Function Documentation	35
6.25.3.1	compare_with_token(List< Token > tokens, Touch other_touch)	35
6.25.3.2	compareTo(Touch other_touch)	35
6.25.3.3	get_key()	36
6.25.3.4	get_pressure()	36
6.25.3.5	get_probability(Window preceeding_window)	36
6.25.3.6	get_timestamp()	36
6.25.3.7	hashCode()	36
6.25.3.8	set_probability(Window preceeding_window, double p)	36
6.25.3.9	toString()	36
6.26	trie.Trie Class Reference	36
6.26.1	Detailed Description	37
6.26.2	Constructor & Destructor Documentation	37
6.26.2.1	Trie()	37
6.26.2.2	Trie(Trie t)	37
6.26.3	Member Function Documentation	37
6.26.3.1	clear()	37
6.26.3.2	get_index_list(String s)	37
6.26.3.3	insertString(String s, int index)	37
6.26.3.4	occurrence_count(String s)	37
6.26.3.5	printSorted(TrieNode node, String s)	37
6.27	trie.TrieList Class Reference	38
6.27.1	Detailed Description	38
6.27.2	Constructor & Destructor Documentation	38
6.27.2.1	TrieList()	38

6.27.2.2	<a href="#">TrieList(TrieList t)</a>	38
6.27.3	<a href="#">Member Function Documentation</a>	38
6.27.3.1	<a href="#">add(Window arg0)</a>	38
6.27.3.2	<a href="#">add(int arg0, Window arg1)</a>	38
6.27.3.3	<a href="#">addAll(Collection&lt;?extends Window &gt; arg0)</a>	39
6.27.3.4	<a href="#">addAll(int arg0, Collection&lt;?extends Window &gt; arg1)</a>	39
6.27.3.5	<a href="#">clear()</a>	39
6.27.3.6	<a href="#">occurrence_count(Window w)</a>	39
6.27.3.7	<a href="#">remove(Object arg0)</a>	39
6.27.3.8	<a href="#">remove(int arg0)</a>	39
6.27.3.9	<a href="#">removeAll(Collection&lt;?&gt; arg0)</a>	39
6.27.3.10	<a href="#">retainAll(Collection&lt;?&gt; arg0)</a>	39
6.27.3.11	<a href="#">set(int arg0, Window arg1)</a>	39
6.27.3.12	<a href="#">set_tokens(List&lt; Token &gt; tokens)</a>	39
6.27.3.13	<a href="#">successor_count(List&lt; Touch &gt; successor_list, Window window, Touch touch)</a>	39
6.28	<a href="#">components.Token.Type Enum Reference</a>	39
6.28.1	<a href="#">Detailed Description</a>	40
6.28.2	<a href="#">Member Data Documentation</a>	40
6.28.2.1	<a href="#">combined</a>	40
6.28.2.2	<a href="#">keycode_mu</a>	40
6.28.2.3	<a href="#">linear</a>	40
6.29	<a href="#">junit.Unit_CompareChainsRank Class Reference</a>	40
6.29.1	<a href="#">Detailed Description</a>	40
6.29.2	<a href="#">Member Function Documentation</a>	40
6.29.2.1	<a href="#">init()</a>	40
6.29.2.2	<a href="#">test_authentication_probability()</a>	40
6.30	<a href="#">junit.Unit_CompleteProbability Class Reference</a>	40
6.30.1	<a href="#">Member Function Documentation</a>	41
6.30.1.1	<a href="#">init()</a>	41
6.30.1.2	<a href="#">test_replica_distribution()</a>	41

6.31 test.UnitCompareChainsRank Class Reference . . . . .	41
6.31.1 Member Function Documentation . . . . .	41
6.31.1.1 init() . . . . .	41
6.31.1.2 test() . . . . .	41
6.31.1.3 test_chain_to_graph() . . . . .	41
6.31.1.4 test_touch_index() . . . . .	41
6.31.1.5 test_touch_window() . . . . .	41
6.32 test.UnitRankCompare Class Reference . . . . .	42
6.32.1 Detailed Description . . . . .	42
6.32.2 Member Function Documentation . . . . .	42
6.32.2.1 init() . . . . .	42
6.32.2.2 test() . . . . .	42
6.32.2.3 test_auth_probability() . . . . .	42
6.32.2.4 test_compare_correct() . . . . .	42
6.33 components.Window Class Reference . . . . .	42
6.33.1 Detailed Description . . . . .	43
6.33.2 Constructor & Destructor Documentation . . . . .	43
6.33.2.1 Window(List< Touch > touches) . . . . .	43
6.33.2.2 Window(Window w) . . . . .	43
6.33.3 Member Function Documentation . . . . .	43
6.33.3.1 compare_with_token(List< Token > tokens, Window other_window) . . . . .	43
6.33.3.2 compareTo(Window other_window) . . . . .	43
6.33.3.3 get_touch_list() . . . . .	43
6.33.3.4 hashCode() . . . . .	44
6.33.3.5 size() . . . . .	44
6.33.3.6 toString() . . . . .	44

<b>7</b>	<b>File Documentation</b>	<b>45</b>
7.1	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/Chain.java File Reference	45
7.2	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/Distribution.java File Reference	45
7.3	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/Token.java File Reference	45
7.4	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/Touch.java File Reference	46
7.5	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/Window.java File Reference	46
7.6	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/data_analysis/Model_↔compare.java File Reference	46
7.7	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/data_analysis/Model_↔compare_thread.java File Reference	47
7.8	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/data_analysis/Statistics.java File Reference	47
7.9	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/Marcov_console_panel.java File Reference	47
7.10	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/Marcov_file_display_↔panel.java File Reference	47
7.11	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/Marcov_frame.java File Reference	48
7.12	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/Marcov_options_panel.java File Reference	48
7.13	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/StartGUI.java File Reference	48
7.14	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/junit/Unit_CompareChains↔Rank.java File Reference	48
7.15	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/junit/Unit_CompleteProbability.java File Reference	49
7.16	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/rank/CompareChainsRank.java File Reference	49
7.17	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/rank/CompleteProbability.java File Reference	49
7.18	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/runtime/ChainBuilder.java File Reference	49
7.19	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/runtime/CompareChains.java File Reference	50

7.20	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/runtime/Operation_thread.java File Reference . . . . .	50
7.21	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/Main.java File Reference .	50
7.22	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/Print_model.java File Ref- erence . . . . .	51
7.23	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/UnitCompareChains↵ Rank.java File Reference . . . . .	51
7.24	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/UnitRankCompare.java File Reference . . . . .	51
7.25	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/Utilities.java File Reference	51
7.26	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/trie/Trie.java File Reference . .	51
7.27	/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/trie/TrieList.java File Reference	52
<b>Index</b>		<b>53</b>





# Chapter 1

## Namespace Index

### 1.1 Packages

Here are the packages with brief descriptions (if available):

<a href="#">components</a>	9
<a href="#">data_analysis</a>	9
<a href="#">gui</a>	9
<a href="#">junit</a>	10
<a href="#">rank</a>	10
<a href="#">runtime</a>	10
<a href="#">test</a>	10
<a href="#">trie</a>	10



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

components.Chain . . . . .	11
runtime.ChainBuilder . . . . .	14
Comparable	
components.Touch . . . . .	34
components.Window . . . . .	42
runtime.ChainBuilder.CompareMethod . . . . .	18
rank.CompleteProbability . . . . .	18
runtime.Operation_thread.Computation . . . . .	19
test.Main.TestFiles.Concentration . . . . .	20
test.Main.TestFiles.Distribution . . . . .	21
components.Distribution . . . . .	22
test.Main . . . . .	23
data_analysis.Model_compare . . . . .	26
test.Main.TestFiles.PressureAmount . . . . .	29
test.Print_model . . . . .	30
Runnable	
data_analysis.Model_compare_thread . . . . .	26
runtime.CompareChains . . . . .	15
rank.CompareChainsRank . . . . .	17
runtime.Operation_thread . . . . .	28
gui.StartGUI . . . . .	30
runtime.ChainBuilder.State . . . . .	30
data_analysis.Statistics . . . . .	31
components.Token . . . . .	32
trie.Trie . . . . .	36
components.Token.Type . . . . .	39
junit.Unit_CompareChainsRank . . . . .	40
junit.Unit_CompleteProbability . . . . .	40
test.UnitCompareChainsRank . . . . .	41
test.UnitRankCompare . . . . .	42
ArrayList	
trie.TrieList . . . . .	38
JFrame	
gui.Marcov_frame . . . . .	25
JPanel	
gui.Marcov_console_panel . . . . .	24
gui.Marcov_file_display_panel . . . . .	24
gui.Marcov_options_panel . . . . .	25



## Chapter 3

# Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">components.Chain</a>	11
<a href="#">runtime.ChainBuilder</a>	14
<a href="#">runtime.CompareChains</a>	
This thread will call the compare method of chain class. The goal is to compare user chain and auth chain and make the result, pass/fail known. Or do something based on pass/fail such as cause the phone to lock	15
<a href="#">rank.CompareChainsRank</a>	17
<a href="#">runtime.ChainBuilder.CompareMethod</a>	18
<a href="#">rank.CompleteProbability</a>	18
<a href="#">runtime.Operation_thread.Computation</a>	19
<a href="#">test.Main.TestFiles.Concentration</a>	20
<a href="#">test.Main.TestFiles.Distribution</a>	21
<a href="#">components.Distribution</a>	22
<a href="#">test.Main</a>	23
<a href="#">gui.Marcov_console_panel</a>	24
<a href="#">gui.Marcov_file_display_panel</a>	24
<a href="#">gui.Marcov_frame</a>	25
<a href="#">gui.Marcov_options_panel</a>	25
<a href="#">data_analysis.Model_compare</a>	26
<a href="#">data_analysis.Model_compare_thread</a>	26
<a href="#">runtime.Operation_thread</a>	28
<a href="#">test.Main.TestFiles.PressureAmount</a>	29
<a href="#">test.Print_model</a>	30
<a href="#">gui.StartGUI</a>	30
<a href="#">runtime.ChainBuilder.State</a>	30
<a href="#">data_analysis.Statistics</a>	31
<a href="#">components.Token</a>	32
<a href="#">components.Touch</a>	
This class represents a touch event	34
<a href="#">trie.Trie</a>	36
<a href="#">trie.TrieList</a>	38
<a href="#">components.Token.Type</a>	
Specify the type of token we want to build	39
<a href="#">junit.Unit_CompareChainsRank</a>	40
<a href="#">junit.Unit_CompleteProbability</a>	40

<a href="#">test.UnitCompareChainsRank</a> . . . . .	41
<a href="#">test.UnitRankCompare</a> . . . . .	42
<a href="#">components.Window</a> This class will store and provide functions for a single window within the model . . . . .	42

## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/Chain.java . . . . .	45
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/Distribution.java . . . . .	45
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/Token.java . . . . .	45
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/Touch.java . . . . .	46
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/Window.java . . . . .	46
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/data_analysis/Model_compare.java	46
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/data_analysis/Model_compare_↵ thread.java . . . . .	47
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/data_analysis/Statistics.java . . . . .	47
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/Marcov_console_panel.java . . .	47
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/Marcov_file_display_panel.java .	47
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/Marcov_frame.java . . . . .	48
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/Marcov_options_panel.java . . .	48
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/StartGUI.java . . . . .	48
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/junit/Unit_CompareChainsRank.java	48
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/junit/Unit_CompleteProbability.java .	49
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/rank/CompareChainsRank.java . . .	49
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/rank/CompleteProbability.java . . .	49
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/runtime/ChainBuilder.java . . . . .	49
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/runtime/CompareChains.java . . . .	50
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/runtime/Operation_thread.java . . .	50
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/Main.java . . . . .	50
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/Print_model.java . . . . .	51
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/UnitCompareChainsRank.java .	51
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/UnitRankCompare.java . . . . .	51
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/Utilities.java . . . . .	51
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/trie/Trie.java . . . . .	51
/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/trie/TrieList.java . . . . .	52





## Chapter 5

# Namespace Documentation

### 5.1 Package components

#### Classes

- class [Chain](#)
- class [Distribution](#)
- class [Token](#)
- class [Touch](#)

*This class represents a touch event.*

- class [Window](#)

*This class will store and provide functions for a single window within the model.*

### 5.2 Package data\_analysis

#### Classes

- class [Model\\_compare](#)
- class [Model\\_compare\\_thread](#)
- class [Statistics](#)

### 5.3 Package gui

#### Classes

- class [Marcov\\_console\\_panel](#)
- class [Marcov\\_file\\_display\\_panel](#)
- class [Marcov\\_frame](#)
- class [Marcov\\_options\\_panel](#)
- class [StartGUI](#)

## 5.4 Package junit

### Classes

- class [Unit\\_CompareChainsRank](#)
- class [Unit\\_CompleteProbability](#)

## 5.5 Package rank

### Classes

- class [CompareChainsRank](#)
- class [CompleteProbability](#)

## 5.6 Package runtime

### Classes

- class [ChainBuilder](#)
- class [CompareChains](#)

*This thread will call the compare method of chain class. The goal is to compare user chain and auth chain and make the result, pass/fail known. Or do something based on pass/fail such as cause the phone to lock.*

- class [Operation\\_thread](#)

## 5.7 Package test

### Classes

- class [Main](#)
- class [Print\\_model](#)
- class [UnitCompareChainsRank](#)
- class [UnitRankCompare](#)

## 5.8 Package trie

### Classes

- class [Trie](#)
- class [TrieList](#)

## Chapter 6

# Class Documentation

### 6.1 components.Chain Class Reference

#### Public Member Functions

- [Chain](#) (int window, int token, int threshold, int model\_size)
- [Chain](#) ([Chain](#) c)  
*copy constructor. New chain object should have the same state as the old with different object references.*
- void [add\\_touch](#) ([Touch](#) touch)
- void [add\\_touch\\_list](#) (List< [Touch](#) > t)
- void [set\\_distribution](#) ([Distribution](#) distribution, List< [Distribution](#) > key\_distribution)
- double [get\\_touch\\_probability](#) ([Window](#) w, [Touch](#) t)  
*returns the probability of a given touch (at the i'th index) based on the model. This will depend on the preceding touches, in [Window](#). A request for one probability will necessarily result in all of the probabilities being computed.*
- [Distribution](#) [get\\_distribution](#) ()  
*returns the distribution of the data as a whole*
- List< [Distribution](#) > [get\\_key\\_distribution](#) ()  
*returns a list of distributions for each key*
- int [get\\_window](#) ()
- int [get\\_token](#) ()
- int [get\\_model\\_size](#) ()
- int [get\\_threshold](#) ()
- void [reset](#) ()  
*resets the object.. this is the same as constructing a new chain, but faster*
- void [compute\\_uncomputed](#) ()  
*computes all uncomputed aspects of the chain*
- double [compare\\_to](#) ([Chain](#) auth\_chain)
- boolean [is\\_touch\\_in\\_key\\_distribution](#) ([Touch](#) touch)  
*returns true if a touch is within 2 sigma for its key distribution*
- List< [Window](#) > [get\\_windows](#) ()  
*handle requests for windows*
- List< [Token](#) > [get\\_tokens](#) ()  
*handle requests for tokens*
- List< [Touch](#) > [get\\_touches](#) ()  
*get a list of all touches in the chain*
- String [toString](#) ()  
*prints out all of the touches in order*
- void [output\\_to\\_csv](#) (String file\_name)  
*NOT USEFUL IN ANDROID. This is used for debugging purposes. Outputs the model to a csv file in a readable format.*

### 6.1.1 Detailed Description

TODO make the chain's compare\_to method be able to update incrementally TODO make sure to use get\_X←XXXXX() instead of the instance variables TODO put windows into a Trie data structure for building model faster TODO anywhere where I need to compare windows, or Touches I need the option to do this with tokens there needs to be a way to set the distribution used for a chain. This is because the authentication chain is evaluated with the distribution of the base chain. compute the windows somewhere. This will be based on the threshold, window, token sizes. This may change distributions? if a touch is thrown out? this class represents the marcov chain. It contains a sequence of touches and a distribution. I avoid doing any processing on touch being added because eventually this will be called on key press in android. Setting it up this way is more flexible to in the sense that processing may be done at any time. caches the result of each computation so it does not have to be repeated.

### 6.1.2 Constructor & Destructor Documentation

6.1.2.1 `components.Chain.Chain ( int window, int token, int threshold, int model_size )`

6.1.2.2 `components.Chain.Chain ( Chain c )`

copy constructor. New chain object should have the same state as the old with differant object references.

### 6.1.3 Member Function Documentation

6.1.3.1 `void components.Chain.add_touch ( Touch touch )`

6.1.3.2 `void components.Chain.add_touch_list ( List< Touch > t )`

6.1.3.3 `double components.Chain.compare_to ( Chain auth_chain )`

returns the percent difference between this chain and auth\_chain. the value returned will be between 0 and 1 0 indicates there is no difference 1 indicates there is a large difference

6.1.3.4 `void components.Chain.compute_uncomputed ( )`

computes all uncomputed aspects of the chain

6.1.3.5 **Distribution** `components.Chain.get_distribution ( )`

returns the distribution of the data as a whole

6.1.3.6 `List<Distribution> components.Chain.get_key_distribution ( )`

returns a list of distributions for each key

6.1.3.7 `int components.Chain.get_model_size ( )`

6.1.3.8 `int components.Chain.get_threshold ( )`

6.1.3.9 `int components.Chain.get_token ( )`

6.1.3.10 `List<Token> components.Chain.get_tokens ( )`

handle requests for tokens

6.1.3.11 `double components.Chain.get_touch_probability ( Window w, Touch t )`

returns the probability of a given touch (at the i'th index) based on the model. This will depend on the preceeding touches, in [Window](#). A request for one probability will necessarily result in all of the probabilities being computed.

6.1.3.12 `List<Touch> components.Chain.get_touches ( )`

get a list of all touches in the chain

6.1.3.13 `int components.Chain.get_window ( )`

6.1.3.14 `List<Window> components.Chain.get_windows ( )`

handle requests for windows

6.1.3.15 `boolean components.Chain.is_touch_in_key_distribution ( Touch touch )`

returns true if a touch is within 2 sigma for it's key distribution

6.1.3.16 `void components.Chain.output_to_csv ( String file_name )`

NOT USEFUL IN ANDROID. This is used for debugging purposes. Outputs the model to a csv file in a readable format.

6.1.3.17 `void components.Chain.reset ( )`

resets the object.. this is the same as constructing a new chain, but faster

6.1.3.18 `void components.Chain.set_distribution ( Distribution distribution, List< Distribution > key_distribution )`

allows distribution to be set. If no distribution is set, the distribution for this chain of touches is computed. NOTE the distribution is not maintained when new touches are added.

### 6.1.3.19 String components.Chain.toString ( )

prints out all of the touches in order

The documentation for this class was generated from the following file:

- [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/components/Chain.java](#)

## 6.2 runtime.ChainBuilder Class Reference

### Classes

- enum [CompareMethod](#)
- enum [State](#)

### Public Member Functions

- [ChainBuilder](#) ( )
- [ChainBuilder](#) (int window, int token, int threshold, int user\_model\_size, int auth\_model\_size)
- void [handle\\_touch](#) ([Touch](#) touch)
- void [authenticate](#) ( )
- [CompareChains](#) [get\\_authenticate\\_thread](#) ( )
- [State](#) [get\\_authenticate\\_state](#) ( )  
*handle requests for the current state of the authentication*
- void [build\\_chain\\_from\\_csv](#) (File file)

### Static Public Member Functions

- static List< [Touch](#) > [parse\\_csv](#) (File file)  
*parse the csv file NOT USEFULL ON ANDROID*

### 6.2.1 Detailed Description

TODO write a HashList class (most likely extends HashMap and implemetns list to store the hash of everything. Replace ArrayList with this class wherever arraylist is used. The other option is to use LinkedHashMap. handles building of the model based on input events. This may not be necessary in android framework, but it will allow a consistant way of building the model across platforms to allow for easier migration to android device. whenever I add a touch, I take

### 6.2.2 Constructor & Destructor Documentation

#### 6.2.2.1 runtime.ChainBuilder.ChainBuilder ( )

#### 6.2.2.2 runtime.ChainBuilder.ChainBuilder ( int window, int token, int threshold, int user\_model\_size, int auth\_model\_size )

allow model size, window, token values to be specified. This is mainly for testing purposes

### 6.2.3 Member Function Documentation

#### 6.2.3.1 void runtime.ChainBuilder.authenticate ( )

allow forced authentication from outside of [ChainBuilder](#). this involves starting the [CompareChains](#). this method starts the authentication

#### 6.2.3.2 void runtime.ChainBuilder.build\_chain\_from\_csv ( File file )

this code will NOT BE USEFULL ON ANDROID. It will build the model from a csv file in the current working directory. It will however utilize the [handle\\_touch\(\)](#) method to add new touches to the chain. It is simply a matter of where the touches are coming from. TODO move this method to another place. it is only by convience that it exists here now.

#### 6.2.3.3 State runtime.ChainBuilder.get\_authenticate\_state ( )

handle requests for the current state of the authentication

#### 6.2.3.4 CompareChains runtime.ChainBuilder.get\_authenticate\_thread ( )

return the thread which is preforming the authentication. This method provides no guarentees about the state of the thread. It may even be null!

#### 6.2.3.5 void runtime.ChainBuilder.handle\_touch ( Touch touch )

this method should be called in some way whenever there is a touch event in android. There should be minimal amounts of processing done here so the input to the device doesn't lag. I don't know by what method percicely this will need to be called in the android souce. It could be another class which simply handles touch events, or from the pre-existing android archetecture.

#### 6.2.3.6 static List<Touch> runtime.ChainBuilder.parse\_csv ( File file ) [static]

parse the csv file NOT USEFULL ON ANDROID

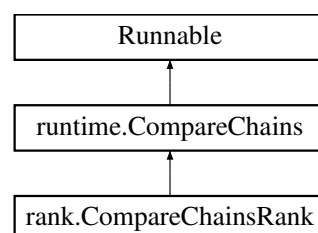
The documentation for this class was generated from the following file:

- [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/runtime/ChainBuilder.java](#)

## 6.3 runtime.CompareChains Class Reference

This thread will call the compare method of chain class. The goal is to compare user chain and auth chain and make the result, pass/fail known. Or do something based on pass/fail such as cause the phone to lock.

Inheritance diagram for runtime.CompareChains:



## Public Member Functions

- [CompareChains](#) ([Chain](#) user\_chain, [Chain](#) auth\_chain)
- void [run](#) ()  
*compare user\_chain and auth\_chain and choose what to do with the result*
- double [get\\_auth\\_probability](#) ()  
*returns the probability with which the*
- boolean [get\\_auth\\_result](#) ()
- boolean [get\\_auth\\_complete](#) ()

## Protected Attributes

- volatile boolean [is\\_authentic](#)
- volatile boolean [complete](#)
- [Chain](#) user\_chain
- [Chain](#) auth\_chain
- volatile double [authentication\\_probability](#)

### 6.3.1 Detailed Description

This thread will call the compare method of chain class. The goal is to compare user chain and auth chain and make the result, pass/fail known. Or do something based on pass/fail such as cause the phone to lock.

### 6.3.2 Constructor & Destructor Documentation

#### 6.3.2.1 runtime.CompareChains.CompareChains ( [Chain](#) user\_chain, [Chain](#) auth\_chain )

will need to make copies of the chains passed in so they do not get updated by something else during the comparison

### 6.3.3 Member Function Documentation

#### 6.3.3.1 boolean runtime.CompareChains.get\_auth\_complete ( )

#### 6.3.3.2 double runtime.CompareChains.get\_auth\_probability ( )

returns the probability with which the

#### 6.3.3.3 boolean runtime.CompareChains.get\_auth\_result ( )

returns the result of the authentication. This method does not provide any guarentees that the compairason has finished yet. If the compairason has not yet finished it will return false;



#### 6.3.3.4 void runtime.CompareChains.run ( )

compare user\_chain and auth\_chain and choose what to do with the result

perform the comparison now that the values are cached in the Chain's

### 6.3.4 Member Data Documentation

#### 6.3.4.1 Chain runtime.CompareChains.auth\_chain [protected]

#### 6.3.4.2 volatile double runtime.CompareChains.authentication\_probability [protected]

#### 6.3.4.3 volatile boolean runtime.CompareChains.complete [protected]

#### 6.3.4.4 volatile boolean runtime.CompareChains.is\_authentic [protected]

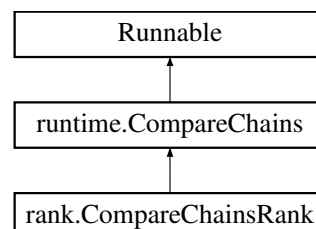
#### 6.3.4.5 Chain runtime.CompareChains.user\_chain [protected]

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/runtime/[CompareChains.java](#)

## 6.4 rank.CompareChainsRank Class Reference

Inheritance diagram for rank.CompareChainsRank:



### Public Member Functions

- [CompareChainsRank](#) (Chain user\_chain, Chain auth\_chain)
- void [run](#) ()

## Additional Inherited Members

### 6.4.1 Constructor & Destructor Documentation

6.4.1.1 `rank.CompareChainsRank.CompareChainsRank ( Chain user_chain, Chain auth_chain )`

### 6.4.2 Member Function Documentation

6.4.2.1 `void rank.CompareChainsRank.run ( )`

overrides the run method to implement the authentication with a page-rank style algorithm.

The documentation for this class was generated from the following file:

- `/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/rank/CompareChainsRank.java`

## 6.5 runtime.ChainBuilder.CompareMethod Enum Reference

### Public Attributes

- `PROBABILITY_VECTOR_DIFFERENCE`

### 6.5.1 Member Data Documentation

6.5.1.1 `runtime.ChainBuilder.CompareMethod.PROBABILITY_VECTOR_DIFFERENCE`

The documentation for this enum was generated from the following file:

- `/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/runtime/ChainBuilder.java`

## 6.6 rank.CompleteProbability Class Reference

### Public Member Functions

- `CompleteProbability (Chain chain)`
- `Chain compute_probability ()`

### 6.6.1 Detailed Description

This class computes probability in a different way from what is contained in the Chain class. This class looks at all of the touches to try to determine the probability that from any given touch, it transitions to another.

this is similar to having a window size of 1?

### Author

element

## 6.6.2 Constructor & Destructor Documentation

### 6.6.2.1 rank.CompleteProbability.CompleteProbability ( Chain *chain* )

## 6.6.3 Member Function Documentation

### 6.6.3.1 Chain rank.CompleteProbability.compute\_probability ( )

make a replica of the chain with a window size of 1 and compute the probability.

#### Returns

replica chain

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/rank/[CompleteProbability.java](#)

## 6.7 runtime.Operation\_thread.Computation Enum Reference

### Public Attributes

- [DISTRIBUTION](#)
- [KEY\\_DISTRIBUTION](#)
- [WINDOW](#)
- [TOKEN](#)
- [PROBABILITY](#)

### 6.7.1 Member Data Documentation

#### 6.7.1.1 runtime.Operation\_thread.Computation.DISTRIBUTION

#### 6.7.1.2 runtime.Operation\_thread.Computation.KEY\_DISTRIBUTION

#### 6.7.1.3 runtime.Operation\_thread.Computation.PROBABILITY

#### 6.7.1.4 runtime.Operation\_thread.Computation.TOKEN

#### 6.7.1.5 runtime.Operation\_thread.Computation.WINDOW

The documentation for this enum was generated from the following file:

- /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/runtime/[Operation\\_thread.java](#)

## 6.8 test.Main.TestFiles.Concentration Enum Reference

### Public Member Functions

- [Concentration](#) (String description, int identifier, double value)
- String [toString](#) ()
- int [get\\_identifier](#) ()
- double [get\\_value](#) ()

### Public Attributes

- [HIGH](#) =("High, [std deviation]", 0, 0)
- [MEDIUM](#) =("Medium, [std deviation]", 1, 0)
- [LOW](#)

### 6.8.1 Constructor & Destructor Documentation

6.8.1.1 `test.Main.TestFiles.Concentration.Concentration ( String description, int identifier, double value )`

### 6.8.2 Member Function Documentation

6.8.2.1 `int test.Main.TestFiles.Concentration.get_identifier ( )`

6.8.2.2 `double test.Main.TestFiles.Concentration.get_value ( )`

6.8.2.3 `String test.Main.TestFiles.Concentration.toString ( )`

### 6.8.3 Member Data Documentation

6.8.3.1 `test.Main.TestFiles.Concentration.HIGH =("High, [std deviation]", 0, 0)`

6.8.3.2 `test.Main.TestFiles.Concentration.LOW`

#### Initial value:

```
=("Low, [std deviation]", 2,
  0)
```

6.8.3.3 `test.Main.TestFiles.Concentration.MEDIUM =("Medium, [std deviation]", 1, 0)`

The documentation for this enum was generated from the following file:

- `/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/Main.java`

## 6.9 test.Main.TestFiles.Distribution Enum Reference

### Public Member Functions

- [Distribution](#) (String description, int identifier, double value)
- String [toString](#) ()
- int [get\\_identifier](#) ()
- double [get\\_value](#) ()

### Public Attributes

- [NORMAL](#) =("Normal, centered about pressure median", 0, 0)
- [ABNORMAL](#)
- [RANDOM](#) =("Random, completly and utterly random", 2, 0)

### 6.9.1 Constructor & Destructor Documentation

6.9.1.1 `test.Main.TestFiles.Distribution.Distribution ( String description, int identifier, double value )`

### 6.9.2 Member Function Documentation

6.9.2.1 `int test.Main.TestFiles.Distribution.get_identifier ( )`

6.9.2.2 `double test.Main.TestFiles.Distribution.get_value ( )`

6.9.2.3 `String test.Main.TestFiles.Distribution.toString ( )`

### 6.9.3 Member Data Documentation

6.9.3.1 `test.Main.TestFiles.Distribution.ABNORMAL`

**Initial value:**

```
= (
    "Abnormal, centered about pressure median, but inverted", 1,
    0)
```

6.9.3.2 `test.Main.TestFiles.Distribution.NORMAL =("Normal, centered about pressure median", 0, 0)`

6.9.3.3 `test.Main.TestFiles.Distribution.RANDOM =("Random, completly and utterly random", 2, 0)`

The documentation for this enum was generated from the following file:

- `/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/test/Main.java`

## 6.10 components.Distribution Class Reference

### Public Member Functions

- [Distribution](#) (List< [Touch](#) > touches)
- [Distribution](#) (List< [Touch](#) > touches, int keycode)  
*this constructor allows a keycode to be associated with the distribution*
- [Distribution](#) ([Distribution](#) d)  
*copy constructor. This exists because computations are done in the constructor. Copying in this way avoids recomputation.*
- void [update](#) (List< [Touch](#) > touches)  
*updates the distribution using a list of touches. This update has nothing to do with the old values in the distribution. It is synonymous to creating a new [Distribution](#) object with this list of touches.*
- double [get\\_min](#) ()
- double [get\\_max](#) ()
- double [get\\_average](#) ()
- double [get\\_standard\\_deviation](#) ()
- int [get\\_keycode](#) ()  
*returns the keycode associated with this distribution. If the distribution does not have an associated keycode, this method will return -1.*
- boolean [equals](#) (Object o)  
*determine if this distribution is exactly equal to another distribution*

### 6.10.1 Detailed Description

TODO make the computations happen at request time, and cache the result so it does not need to be recomputed. Or leave it as is.... as distribution objects are only created as needed in the rest of the code. this class knows how to calculate the distribution of a list of touches

### 6.10.2 Constructor & Destructor Documentation

#### 6.10.2.1 components.Distribution.Distribution ( List< [Touch](#) > touches )

#### 6.10.2.2 components.Distribution.Distribution ( List< [Touch](#) > touches, int keycode )

this constructor allows a keycode to be associated with the distribution

#### 6.10.2.3 components.Distribution.Distribution ( [Distribution](#) d )

copy constructor. This exists because computations are done in the constructor. Copying in this way avoids recomputation.

### 6.10.3 Member Function Documentation

#### 6.10.3.1 boolean components.Distribution.equals ( Object o )

determine if this distribution is exactly equal to another distribution

6.10.3.2 double components.Distribution.get\_average ( )

6.10.3.3 int components.Distribution.get\_keycode ( )

returns the keycode associated with this distribution. If the distribution does not have an associated keycode, this method will return -1.

6.10.3.4 double components.Distribution.get\_max ( )

6.10.3.5 double components.Distribution.get\_min ( )

6.10.3.6 double components.Distribution.get\_standard\_deviation ( )

6.10.3.7 void components.Distribution.update ( List< Touch > touches )

updates the distribution using a list of touches. This update has nothing to do with the old values in the distribution. It is synonymous to creating a new [Distribution](#) object with this list of touches.

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/components/[Distribution.java](#)

## 6.11 test.Main Class Reference

### Static Public Member Functions

- static void [main](#) (String args[])

### 6.11.1 Detailed Description

TODO generate a csv file for testing This class is used to test that the model is being built correctly. Also tested is the model compairason. and various classes used in model creating. The idea is to print out the tests which fail. This class should have to do no actual work if the program is designed well.

### 6.11.2 Member Function Documentation

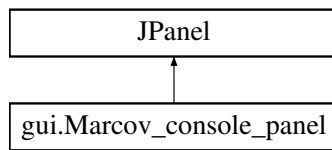
6.11.2.1 static void test.Main.main ( String args[] ) [static]

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/test/[Main.java](#)

## 6.12 gui.Marcov\_console\_panel Class Reference

Inheritance diagram for gui.Marcov\_console\_panel:



### Public Member Functions

- [Marcov\\_console\\_panel \(\)](#)

### 6.12.1 Constructor & Destructor Documentation

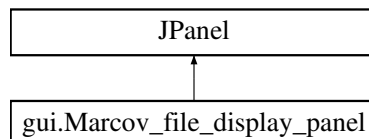
#### 6.12.1.1 gui.Marcov\_console\_panel.Marcov\_console\_panel ( )

The documentation for this class was generated from the following file:

- [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/gui/Marcov\\_console\\_panel.java](#)

## 6.13 gui.Marcov\_file\_display\_panel Class Reference

Inheritance diagram for gui.Marcov\_file\_display\_panel:



### Public Member Functions

- [Marcov\\_file\\_display\\_panel \(\)](#)

### 6.13.1 Constructor & Destructor Documentation

#### 6.13.1.1 gui.Marcov\_file\_display\_panel.Marcov\_file\_display\_panel ( )

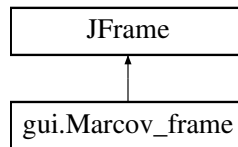
The documentation for this class was generated from the following file:

- [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/gui/Marcov\\_file\\_display\\_panel.java](#)



## 6.14 gui.Marcov\_frame Class Reference

Inheritance diagram for gui.Marcov\_frame:



### Public Member Functions

- [Marcov\\_frame](#) ()
- void [close](#) ()

### 6.14.1 Constructor & Destructor Documentation

6.14.1.1 `gui.Marcov_frame.Marcov_frame ( )`

### 6.14.2 Member Function Documentation

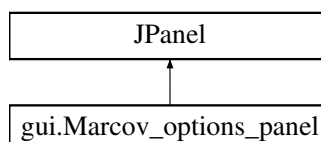
6.14.2.1 `void gui.Marcov_frame.close ( )`

The documentation for this class was generated from the following file:

- `/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/Marcov\_frame.java`

## 6.15 gui.Marcov\_options\_panel Class Reference

Inheritance diagram for gui.Marcov\_options\_panel:



### Public Member Functions

- [Marcov\\_options\\_panel](#) ()

### 6.15.1 Constructor & Destructor Documentation

#### 6.15.1.1 `gui.Marcov_options_panel.Marcov_options_panel ( )`

The documentation for this class was generated from the following file:

- `/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/gui/Marcov_options_panel.java`

## 6.16 `data_analysis.Model_compare` Class Reference

### Static Public Member Functions

- static void `main` (`String[]` args)

#### 6.16.1 Detailed Description

The purpose of this class is to test out the model compare process on data that has been collected The data to used will be contained in the `data_sets` folder input: `data_sets` folder output: `model_compare_output.txt`

#### 6.16.2 Member Function Documentation

##### 6.16.2.1 `static void data_analysis.Model_compare.main ( String[] args )` `[static]`

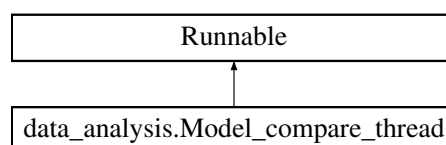
create a number of tests with different parameters

The documentation for this class was generated from the following file:

- `/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/data_analysis/Model_compare.java`

## 6.17 `data_analysis.Model_compare_thread` Class Reference

Inheritance diagram for `data_analysis.Model_compare_thread`:



## Public Member Functions

- [Model\\_compare\\_thread](#) (String base\_data\_path, String auth\_data\_path, int base\_model\_size, int auth\_model\_size, int window\_size, int token\_size, int threshold)  
*constructor, allowing user to set different propperties of the model compairason for testing*
- void [run](#) ()
- String [get\\_base\\_data\\_path](#) ()
- String [get\\_auth\\_data\\_path](#) ()
- int [get\\_window\\_size](#) ()
- int [get\\_token\\_size](#) ()
- int [get\\_threshold](#) ()
- int [get\\_base\\_model\\_size](#) ()
- int [get\\_auth\\_model\\_size](#) ()
- List< Double > [get\\_auth\\_probability\\_list](#) ()

## Public Attributes

- double [max\\_authentication\\_probability](#)
- double [min\\_authentication\\_probability](#)
- double [average\\_authentication\\_probability](#)

### 6.17.1 Detailed Description

this thread allows the preforming of a test compairason. when the compairason is finished, an instance variable will be set indicating different results.

### 6.17.2 Constructor & Destructor Documentation

- 6.17.2.1 `data_analysis.Model_compare_thread.Model_compare_thread ( String base_data_path, String auth_data_path, int base_model_size, int auth_model_size, int window_size, int token_size, int threshold )`

constructor, allowing user to set different propperties of the model compairason for testing

### 6.17.3 Member Function Documentation

- 6.17.3.1 `String data_analysis.Model_compare_thread.get_auth_data_path ( )`

- 6.17.3.2 `int data_analysis.Model_compare_thread.get_auth_model_size ( )`

- 6.17.3.3 `List<Double> data_analysis.Model_compare_thread.get_auth_probability_list ( )`

- 6.17.3.4 `String data_analysis.Model_compare_thread.get_base_data_path ( )`

- 6.17.3.5 `int data_analysis.Model_compare_thread.get_base_model_size ( )`

- 6.17.3.6 `int data_analysis.Model_compare_thread.get_threshold ( )`

6.17.3.7 `int data_analysis.Model_compare_thread.get_token_size ( )`

6.17.3.8 `int data_analysis.Model_compare_thread.get_window_size ( )`

6.17.3.9 `void data_analysis.Model_compare_thread.run ( )`

#### 6.17.4 Member Data Documentation

6.17.4.1 `double data_analysis.Model_compare_thread.average_authentication_probability`

6.17.4.2 `double data_analysis.Model_compare_thread.max_authentication_probability`

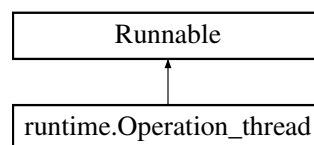
6.17.4.3 `double data_analysis.Model_compare_thread.min_authentication_probability`

The documentation for this class was generated from the following file:

- [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/data\\_analysis/Model\\_compare\\_thread.java](#)

### 6.18 runtime.Operation\_thread Class Reference

Inheritance diagram for runtime.Operation\_thread:



#### Classes

- enum [Computation](#)

#### Public Member Functions

- [Operation\\_thread](#) ([Chain](#) chain, [Computation](#) computation)
- void [run](#) ()

#### 6.18.1 Constructor & Destructor Documentation

6.18.1.1 `runtime.Operation_thread.Operation_thread ( Chain chain, Computation computation )`

#### 6.18.2 Member Function Documentation

6.18.2.1 `void runtime.Operation_thread.run ( )`

The documentation for this class was generated from the following file:

- [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/runtime/Operation\\_thread.java](#)

## 6.19 test.Main.TestFiles.PressureAmount Enum Reference

### Public Member Functions

- [PressureAmount](#) (String description, int identifier, double value)
- String [toString](#) ()
- int [get\\_identifier](#) ()
- double [get\\_value](#) ()

### Public Attributes

- [HIGH](#) =("High pressure, 0.75", 0, .75)
- [MEDIUM](#) =("Medium Pressure, 0.5", 1, .5)
- [LOW](#)

### 6.19.1 Constructor & Destructor Documentation

6.19.1.1 test.Main.TestFiles.PressureAmount.PressureAmount ( String *description*, int *identifier*, double *value* )

### 6.19.2 Member Function Documentation

6.19.2.1 int test.Main.TestFiles.PressureAmount.get\_identifier ( )

6.19.2.2 double test.Main.TestFiles.PressureAmount.get\_value ( )

6.19.2.3 String test.Main.TestFiles.PressureAmount.toString ( )

### 6.19.3 Member Data Documentation

6.19.3.1 test.Main.TestFiles.PressureAmount.HIGH =("High pressure, 0.75", 0, .75)

6.19.3.2 test.Main.TestFiles.PressureAmount.LOW

#### Initial value:

```
=("Low Pressure, 0.25", 2,  
  .25)
```

6.19.3.3 test.Main.TestFiles.PressureAmount.MEDIUM =("Medium Pressure, 0.5", 1, .5)

The documentation for this enum was generated from the following file:

- /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/test/[Main.java](#)

## 6.20 test.Print\_model Class Reference

### Static Public Member Functions

- static void [main](#) (String[] args)

### 6.20.1 Member Function Documentation

6.20.1.1 static void test.Print\_model.main ( String[] *args* ) [static]

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/test/[Print\\_model.java](#)

## 6.21 gui.StartGUI Class Reference

### Static Public Member Functions

- static void [main](#) (String[] args)
- static void [exit](#) ()  
*causes the frame to close*

### 6.21.1 Member Function Documentation

6.21.1.1 static void gui.StartGUI.exit ( ) [static]

causes the frame to close

6.21.1.2 static void gui.StartGUI.main ( String[] *args* ) [static]

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/gui/[StartGUI.java](#)

## 6.22 runtime.ChainBuilder.State Enum Reference

### Public Attributes

- [IN\\_PROGRESS](#)
- [SUCCESS](#)

### 6.22.1 Member Data Documentation

6.22.1.1 runtime.ChainBuilder.State.IN\_PROGRESS

6.22.1.2 runtime.ChainBuilder.State.SUCCESS

The documentation for this enum was generated from the following file:

- /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/runtime/[ChainBuilder.java](#)

## 6.23 data\_analysis.Statistics Class Reference

### Static Public Member Functions

- static void [main](#) (String args[])
- static double [false\\_positive\\_percentage](#) (double authentication\_percentage, List< Double > should\_authenticate\_percentages, List< Double > should\_not\_authenticate\_percentages)
- static double [false\\_negative\\_percentage](#) (double authentication\_percentage, List< Double > should\_authenticate\_percentages, List< Double > should\_not\_authenticate\_percentages)
- static double [best\\_authentication\\_percentage](#) (List< Double > should\_authenticate\_percentages, List< Double > should\_not\_authenticate\_percentages)
- static double [minimize\\_false\\_positive\\_authentication\\_percentage](#) (List< Double > should\_authenticate\_percentages, List< Double > should\_not\_authenticate\_percentages)
- static double [equal\\_false\\_positive\\_negative\\_authentication\\_percentage](#) (List< Double > should\_authenticate\_percentages, List< Double > should\_not\_authenticate\_percentages)
- static double [authentication\\_accuracy](#) (double authentication\_percentage, List< Double > should\_authenticate\_percentages, List< Double > should\_not\_authenticate\_percentages)

### 6.23.1 Member Function Documentation

- 6.23.1.1 static double data\_analysis.Statistics.authentication\_accuracy ( double authentication\_percentage, List< Double > should\_authenticate\_percentages, List< Double > should\_not\_authenticate\_percentages ) [static]
- 6.23.1.2 static double data\_analysis.Statistics.best\_authentication\_percentage ( List< Double > should\_authenticate\_percentages, List< Double > should\_not\_authenticate\_percentages ) [static]
- 6.23.1.3 static double data\_analysis.Statistics.equal\_false\_positive\_negative\_authentication\_percentage ( List< Double > should\_authenticate\_percentages, List< Double > should\_not\_authenticate\_percentages ) [static]
- 6.23.1.4 static double data\_analysis.Statistics.false\_negative\_percentage ( double authentication\_percentage, List< Double > should\_authenticate\_percentages, List< Double > should\_not\_authenticate\_percentages ) [static]
- 6.23.1.5 static double data\_analysis.Statistics.false\_positive\_percentage ( double authentication\_percentage, List< Double > should\_authenticate\_percentages, List< Double > should\_not\_authenticate\_percentages ) [static]
- 6.23.1.6 static void data\_analysis.Statistics.main ( String args[] ) [static]
- 6.23.1.7 static double data\_analysis.Statistics.minimize\_false\_positive\_authentication\_percentage ( List< Double > should\_authenticate\_percentages, List< Double > should\_not\_authenticate\_percentages ) [static]

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/data\_analysis/[Statistics.java](#)

## 6.24 components.Token Class Reference

### Classes

- enum [Type](#)  
*specify the type of token we want to build*

### Public Member Functions

- [Token](#) ([Distribution](#) distribution, int total\_tokens, int token\_index, double standard\_deviations, [Type](#) type)  
*allow tokens to be created making each touch mu for its keycode, or in a linear fashion over the distribution for the keycode*
- [Token](#) ([Distribution](#) distribution, int total\_tokens, int token\_index, [Type](#) type)  
*allow for creation of tokens over a distribution*
- [Token](#) (double range\_min, double range\_max, int total\_tokens, int token\_index, [Type](#) type)
- boolean [contains](#) ([Touch](#) touch)
- boolean [is\\_high\\_wildcard](#) ([Touch](#) touch)  
*determine if a touch is a high\_wildcard*
- boolean [is\\_low\\_wildcard](#) ([Touch](#) touch)  
*determine if a touch is a low wildcard*
- void [increment\\_high\\_wildcards](#) ()  
*adds the the number of high wildcards*
- void [increment\\_low\\_wildcards](#) ()  
*adds the the number of high wildcards*
- int [get\\_total\\_wildcards](#) ()  
*returns the total number of wildcards*
- int [get\\_acceptable\\_wildcards](#) (int total\_items)
- double [get\\_min](#) ()  
*returns true if there are more than an acceptable number of wildcards*
- double [get\\_max](#) ()  
*return the maximum*
- boolean [equals](#) (Object o\_t)  
*compares This token to another to determine if they are the same*

### 6.24.1 Detailed Description

This class represents a token within the model. Essentially this is a range of values. A touch is defined to be within a token if the pressure value of the touch falls within this range. This class is designed to abstract away the clustering algorithm. This makes the rest of the code far simpler to think about Something to look at in the future may be a clustering algorithm that is not equally distributed

### 6.24.2 Constructor & Destructor Documentation

#### 6.24.2.1 components.Token.Token ( [Distribution](#) distribution, int total\_tokens, int token\_index, double standard\_deviations, [Type](#) type )

allow tokens to be created making each touch mu for its keycode, or in a linear fashion over the distribution for the keycode

allow for creation of tokens with-in some number of standard deviations of a distribution



6.24.2.2 components.Token.Token ( *Distribution* *distribution*, int *total\_tokens*, int *token\_index*, *Type* *type* )

allow for creation of tokens over a distribution

6.24.2.3 components.Token.Token ( double *range\_min*, double *range\_max*, int *total\_tokens*, int *token\_index*, *Type* *type* )

Implemented in the constructor is the clustering algorithm. This determines how to split up the range into a number of tokens.

*range\_min*, minimum of the token range *range\_max*, maximum of the token range *total\_tokens* total number of tokens to split range into *token\_index*, integer between 0 and *total\_tokens*-1 indicating into which range this token falls

## 6.24.3 Member Function Documentation

6.24.3.1 boolean components.Token.contains ( *Touch* *touch* )

determines if a touch is within this token based on its pressure value this will return true if a touches pressure equals max or min, so if max of one token is min of another token, both will return true

6.24.3.2 boolean components.Token.equals ( *Object* *o\_t* )

compares This token to another to determine if they are the same

6.24.3.3 int components.Token.get\_acceptable\_wildcards ( int *total\_items* )

returns the acceptable number of wildcards

## Parameters

<i>total_items</i>	is the total number of items in the distribution
--------------------	--

## 6.24.3.4 double components.Token.get\_max ( )

return the maximum

## 6.24.3.5 double components.Token.get\_min ( )

returns true if there are more than an acceptable number of wildcards

return minimum

#### 6.24.3.6 `int components.Token.get_total_wildcards ( )`

returns the total number of wildcards

#### 6.24.3.7 `void components.Token.increment_high_wildcards ( )`

adds the the number of high wildcards

#### 6.24.3.8 `void components.Token.increment_low_wildcards ( )`

adds the the number of high wildcards

#### 6.24.3.9 `boolean components.Token.is_high_wildcard ( Touch touch )`

determine if a touch is a high\_wildcard

#### 6.24.3.10 `boolean components.Token.is_low_wildcard ( Touch touch )`

determine if a touch is a low wildcard

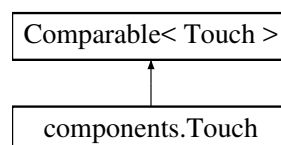
The documentation for this class was generated from the following file:

- `/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/Token.java`

## 6.25 components.Touch Class Reference

This class represents a touch event.

Inheritance diagram for components.Touch:



## Public Member Functions

- [Touch](#) (int keycode, double pressure, long timestamp)
- [Touch](#) ([Touch](#) t)  
*copy constructor*
- void [set\\_probability](#) ([Window](#) preceeding\_window, double p)  
*sets the probability that this touch succeeds a given sequence. Reccord the sequence and the probability*
- double [get\\_probability](#) ([Window](#) preceeding\_window)  
*returns the probability of the touch occurring after a given window w. If the window does not exist return (TODO) currently returning 0*
- double [get\\_pressure](#) ()
- int [get\\_key](#) ()
- long [get\\_timestamp](#) ()
- boolean [compare\\_with\\_token](#) (List< [Token](#) > tokens, [Touch](#) other\_touch)
- int [hashCode](#) ()  
*implement hash function for the touch class*
- int [compareTo](#) ([Touch](#) other\_touch)  
*compare touches to one another. return negative if this touch is less than other\_touch*
- String [toString](#) ()

### 6.25.1 Detailed Description

This class represents a touch event.

### 6.25.2 Constructor & Destructor Documentation

6.25.2.1 `components.Touch.Touch ( int keycode, double pressure, long timestamp )`

6.25.2.2 `components.Touch.Touch ( Touch t )`

copy constructor

### 6.25.3 Member Function Documentation

6.25.3.1 `boolean components.Touch.compare_with_token ( List< Token > tokens, Touch other_touch )`

compares the touches with the given token list. this function will return true if the touches are contained within the same token

6.25.3.2 `int components.Touch.compareTo ( Touch other_touch )`

compare touches to one another. return negative if this touch is less than other\_touch

6.25.3.3 int components.Touch.get\_key ( )

6.25.3.4 double components.Touch.get\_pressure ( )

6.25.3.5 double components.Touch.get\_probability ( Window preceeding\_window )

returns the probability of the touch occurring after a given window w. If the window does not exist return (TODO) currently returning 0

6.25.3.6 long components.Touch.get\_timestamp ( )

6.25.3.7 int components.Touch.hashCode ( )

implement hash function for the touch class

6.25.3.8 void components.Touch.set\_probability ( Window preceeding\_window, double p )

sets the probability that this touch succeeds a given sequence. Reccord the sequence and the probability

6.25.3.9 String components.Touch.toString ( )

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/components/[Touch.java](#)

## 6.26 trie.Trie Class Reference

### Classes

- class **TrieNode**

### Public Member Functions

- [Trie](#) ()  
*sets up the tree so that everything will be added to trienode root?*
- [Trie](#) (Trie t)  
*creates a copy trie*
- void [clear](#) ()  
*removes all elements from the trie*
- void [insertString](#) (String s, int index)
- int [occurrence\\_count](#) (String s)  
*retrieves the number of occurrences of a given string in the tree*
- List< Integer > [get\\_index\\_list](#) (String s)  
*returns a list of indexes containing the given window*
- void [printSorted](#) (TrieNode node, String s)  
*prints the elements in a sorted order*

### 6.26.1 Detailed Description

Credit for portions of this implementation to: <https://sites.google.com/site/indy256/algo/trie>

### 6.26.2 Constructor & Destructor Documentation

#### 6.26.2.1 `trie.Trie.Trie ( )`

sets up the tree so that everything will be added to trienode root?

#### 6.26.2.2 `trie.Trie.Trie ( Trie t )`

creates a copy trie

### 6.26.3 Member Function Documentation

#### 6.26.3.1 `void trie.Trie.clear ( )`

removes all elements from the trie

#### 6.26.3.2 `List<Integer> trie.Trie.get_index_list ( String s )`

returns a list of indexes containing the given window

#### 6.26.3.3 `void trie.Trie.insertString ( String s, int index )`

#### 6.26.3.4 `int trie.Trie.occurrence_count ( String s )`

retrieves the number of occurrences of a given string in the tree

#### 6.26.3.5 `void trie.Trie.printSorted ( TrieNode node, String s )`

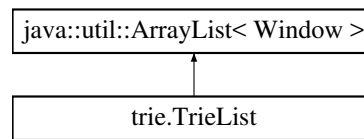
prints the elements in a sorted order

The documentation for this class was generated from the following file:

- `/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/trie/Trie.java`

## 6.27 trie.TrieList Class Reference

Inheritance diagram for trie.TrieList:



### Public Member Functions

- [TrieList](#) ()
- [TrieList](#) ([TrieList](#) t)
- boolean [add](#) ([Window](#) arg0)
- void [add](#) (int arg0, [Window](#) arg1)
- boolean [addAll](#) (Collection<? extends [Window](#) > arg0)
- boolean [addAll](#) (int arg0, Collection<? extends [Window](#) > arg1)
- void [clear](#) ()
- boolean [remove](#) (Object arg0)
- [Window](#) [remove](#) (int arg0)
- boolean [removeAll](#) (Collection<?> arg0)
- boolean [retainAll](#) (Collection<?> arg0)
- [Window](#) [set](#) (int arg0, [Window](#) arg1)
- int [successor\\_count](#) (List< [Touch](#) > successor\_list, [Window](#) window, [Touch](#) touch)
  - counts the number of times a given touch comes after a given window. in the given window, succesors list*
- int [occurrence\\_count](#) ([Window](#) w)
- void [set\\_tokens](#) (List< [Token](#) > tokens)
  - sets the tokens that will be used when encoding the window*

### 6.27.1 Detailed Description

TODO Eventually this will be implemented as a prefix tree. This will greatly speed up many of the operations causing the calculation of the probabilities to be slow. right now it is fine to have the backing be an arraylist overrided methods are any that remove, modify, or add to the arraylist. these methods will also change the prefix tree

### 6.27.2 Constructor & Destructor Documentation

6.27.2.1 [trie.TrieList.TrieList](#) ( )

6.27.2.2 [trie.TrieList.TrieList](#) ( [TrieList](#) t )

### 6.27.3 Member Function Documentation

6.27.3.1 boolean [trie.TrieList.add](#) ( [Window](#) arg0 )

6.27.3.2 void [trie.TrieList.add](#) ( int arg0, [Window](#) arg1 )

6.27.3.3 `boolean trie.TrieList.addAll ( Collection<?extends Window > arg0 )`

6.27.3.4 `boolean trie.TrieList.addAll ( int arg0, Collection<?extends Window > arg1 )`

6.27.3.5 `void trie.TrieList.clear ( )`

6.27.3.6 `int trie.TrieList.occurrence_count ( Window w )`

return the number of occurrences of w in window\_list TODO I think this method needs to be faster. Storing windows in a prefix tree would allow for this

6.27.3.7 `boolean trie.TrieList.remove ( Object arg0 )`

6.27.3.8 `Window trie.TrieList.remove ( int arg0 )`

6.27.3.9 `boolean trie.TrieList.removeAll ( Collection<?> arg0 )`

6.27.3.10 `boolean trie.TrieList.retainAll ( Collection<?> arg0 )`

6.27.3.11 `Window trie.TrieList.set ( int arg0, Window arg1 )`

6.27.3.12 `void trie.TrieList.set_tokens ( List< Token > tokens )`

sets the tokens that will be used when encoding the window

6.27.3.13 `int trie.TrieList.successor_count ( List< Touch > successor_list, Window window, Touch touch )`

counts the number of times a given touch comes after a given window. in the given window, successors list

The documentation for this class was generated from the following file:

- `/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/trie/TrieList.java`

## 6.28 components.Token.Type Enum Reference

specify the type of token we want to build

### Public Attributes

- `linear`
- `keycode_mu`
- `combined`

### 6.28.1 Detailed Description

specify the type of token we want to build

### 6.28.2 Member Data Documentation

6.28.2.1 `components.Token.Type.combined`

6.28.2.2 `components.Token.Type.keycode_mu`

6.28.2.3 `components.Token.Type.linear`

The documentation for this enum was generated from the following file:

- `/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/Token.java`

## 6.29 junit.Unit\_CompareChainsRank Class Reference

### Public Member Functions

- void [init](#) ()
- void [test\\_authentication\\_probability](#) ()

### 6.29.1 Detailed Description

goal is to test compare chains rank functionality

### 6.29.2 Member Function Documentation

6.29.2.1 void `junit.Unit_CompareChainsRank.init` ( )

6.29.2.2 void `junit.Unit_CompareChainsRank.test_authentication_probability` ( )

The documentation for this class was generated from the following file:

- `/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/junit/Unit\_CompareChainsRank.java`

## 6.30 junit.Unit\_CompleteProbability Class Reference

### Public Member Functions

- void [init](#) ()
- void [test\\_replica\\_distribution](#) ()



### 6.30.1 Member Function Documentation

6.30.1.1 void junit.Unit\_CompleteProbability.init ( )

6.30.1.2 void junit.Unit\_CompleteProbability.test\_replica\_distribution ( )

test different properties of replica chain to see if this works as expected. Replica chain should contain the probabilities for when window is equal to 1.

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/junit/[Unit\\_CompleteProbability.java](#)

## 6.31 test.UnitCompareChainsRank Class Reference

### Public Member Functions

- void [init](#) ()
- void [test\\_chain\\_to\\_graph](#) ()
- void [test\\_touch\\_index](#) ()
- void [test\\_touch\\_window](#) ()
- void [test](#) ()

### 6.31.1 Member Function Documentation

6.31.1.1 void test.UnitCompareChainsRank.init ( )

6.31.1.2 void test.UnitCompareChainsRank.test ( )

example test

6.31.1.3 void test.UnitCompareChainsRank.test\_chain\_to\_graph ( )

tests chain\_to\_graph method to make sure the chain is converted to a StateGraph correctly.

6.31.1.4 void test.UnitCompareChainsRank.test\_touch\_index ( )

make sure touch index returns the correct index in the list

6.31.1.5 void test.UnitCompareChainsRank.test\_touch\_window ( )

make sure the touch\_window() returns the correct window in chain.

The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/test/[UnitCompareChainsRank.java](#)

## 6.32 test.UnitRankCompare Class Reference

### Public Member Functions

- void [init](#) ()
- void [test\\_compare\\_correct](#) ()
- void [test\\_auth\\_probability](#) ()
- void [test](#) ()

### 6.32.1 Detailed Description

Test the compairason with ranks

Author

element

### 6.32.2 Member Function Documentation

6.32.2.1 void test.UnitRankCompare.init ( )

6.32.2.2 void test.UnitRankCompare.test ( )

example test

6.32.2.3 void test.UnitRankCompare.test\_auth\_probability ( )

check the test that the compare vectors are correct

6.32.2.4 void test.UnitRankCompare.test\_compare\_correct ( )

checks that the probabilities are correct

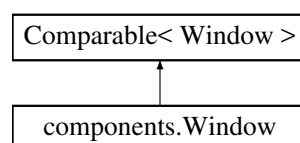
The documentation for this class was generated from the following file:

- /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/test/[UnitRankCompare.java](#)

## 6.33 components.Window Class Reference

This class will store and provide functions for a single window within the model.

Inheritance diagram for components.Window:



## Public Member Functions

- [Window](#) (List< [Touch](#) > touches)
- [Window](#) ([Window](#) w)  
*copy constructor*
- boolean [compare\\_with\\_token](#) (List< [Token](#) > tokens, [Window](#) other\_window)
- int [size](#) ()  
*returns the number of touches in the window*
- List< [Touch](#) > [get\\_touch\\_list](#) ()  
*returns the window in the form of a touch list*
- int [hashCode](#) ()  
*implement a hash function which returns the hash of the current window*
- int [compareTo](#) ([Window](#) other\_window)  
*compare this window to another window. Return negative if this window is less than the other window. Comparason is based on touches' pressure. Returns 0 if they are equal.*
- String [toString](#) ()

### 6.33.1 Detailed Description

This class will store and provide functions for a single window within the model.

### 6.33.2 Constructor & Destructor Documentation

6.33.2.1 `components.Window.Window ( List< Touch > touches )`

6.33.2.2 `components.Window.Window ( Window w )`

copy constructor

### 6.33.3 Member Function Documentation

6.33.3.1 `boolean components.Window.compare_with_token ( List< Token > tokens, Window other_window )`

used for compairason of windows with a given token set. return true if this window is equal to auth window.

6.33.3.2 `int components.Window.compareTo ( Window other_window )`

compare this window to another window. Return negative if this window is less than the other window. Comparason is based on touches' pressure. Returns 0 if they are equal.

6.33.3.3 `List<Touch> components.Window.get_touch_list ( )`

returns the window in the form of a touch list

**6.33.3.4** `int components.Window.hashCode ( )`

implement a hash function which returns the hash of the current window

**6.33.3.5** `int components.Window.size ( )`

returns the number of touches in the window

**6.33.3.6** `String components.Window.toString ( )`

The documentation for this class was generated from the following file:

- `/home/element/PUF/Keyboard/java_scripts/java_marcov_model/src/components/Window.java`

## Chapter 7

# File Documentation

### 7.1 /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/components/↵ Chain.java File Reference

#### Classes

- class [components.Chain](#)

#### Packages

- package [components](#)

### 7.2 /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/components/↵ Distribution.java File Reference

#### Classes

- class [components.Distribution](#)

#### Packages

- package [components](#)

### 7.3 /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/components/↵ Token.java File Reference

#### Classes

- class [components.Token](#)
- enum [components.Token.Type](#)

*specify the type of token we want to build*

## Packages

- package [components](#)

## 7.4 /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/components/↵ Touch.java File Reference

### Classes

- class [components.Touch](#)  
*This class represents a touch event.*

## Packages

- package [components](#)

## 7.5 /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/components/↵ Window.java File Reference

### Classes

- class [components.Window](#)  
*This class will store and provide functions for a single window within the model.*

## Packages

- package [components](#)

## 7.6 /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/data\_analysis/↵ Model\_compare.java File Reference

### Classes

- class [data\\_analysis.Model\\_compare](#)

## Packages

- package [data\\_analysis](#)

**7.7** [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/data\\_analysis/Model\\_compare\\_thread.java](#) File Reference

**7.7** [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/data\\_analysis/Model\\_compare\\_thread.java](#) File Reference 47

## Classes

- class [data\\_analysis.Model\\_compare\\_thread](#)

## Packages

- package [data\\_analysis](#)

**7.8** [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/data\\_analysis/Statistics.java](#) File Reference

## Classes

- class [data\\_analysis.Statistics](#)

## Packages

- package [data\\_analysis](#)

**7.9** [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/gui/Marcov\\_console\\_panel.java](#) File Reference

## Classes

- class [gui.Marcov\\_console\\_panel](#)

## Packages

- package [gui](#)

**7.10** [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/gui/Marcov\\_file\\_display\\_panel.java](#) File Reference

## Classes

- class [gui.Marcov\\_file\\_display\\_panel](#)

## Packages

- package [gui](#)

### 7.11 [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/gui/Marcov\\_frame.java](#) File Reference

## Classes

- class [gui.Marcov\\_frame](#)

## Packages

- package [gui](#)

### 7.12 [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/gui/Marcov\\_options\\_panel.java](#) File Reference

## Classes

- class [gui.Marcov\\_options\\_panel](#)

## Packages

- package [gui](#)

### 7.13 [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/gui/StartGUI.java](#) File Reference

## Classes

- class [gui.StartGUI](#)

## Packages

- package [gui](#)

### 7.14 [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/junit/Unit\\_CompareChainsRank.java](#) File Reference

## Classes

- class [junit.Unit\\_CompareChainsRank](#)



---

Packages

- package [junit](#)

## 7.15 [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/junit/Unit\\_CompleteProbability.java](#) File Reference

## Classes

- class [junit.Unit\\_CompleteProbability](#)

## Packages

- package [junit](#)

## 7.16 [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/rank/CompareChainsRank.java](#) File Reference

## Classes

- class [rank.CompareChainsRank](#)

## Packages

- package [rank](#)

## 7.17 [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/rank/CompleteProbability.java](#) File Reference

## Classes

- class [rank.CompleteProbability](#)

## Packages

- package [rank](#)

## 7.18 [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/runtime/ChainBuilder.java](#) File Reference

## Classes

- class [runtime.ChainBuilder](#)
- enum [runtime.ChainBuilder.State](#)
- enum [runtime.ChainBuilder.CompareMethod](#)

## Packages

- package [runtime](#)

## 7.19 [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/runtime/CompareChains.java](#) File Reference

## Classes

- class [runtime.CompareChains](#)

*This thread will call the compare method of chain class. The goal is to compare user chain and auth chain and make the result, pass/fail known. Or do something based on pass/fail such as cause the phone to lock.*

## Packages

- package [runtime](#)

## 7.20 [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/runtime/Operation\\_thread.java](#) File Reference

## Classes

- class [runtime.Operation\\_thread](#)
- enum [runtime.Operation\\_thread.Computation](#)

## Packages

- package [runtime](#)

## 7.21 [/home/element/PUF/Keyboard/java\\_scripts/java\\_marcov\\_model/src/test/Main.java](#) File Reference

## Classes

- class [test.Main](#)
- enum [test.Main.TestFiles.PressureAmount](#)
- enum [test.Main.TestFiles.Distribution](#)
- enum [test.Main.TestFiles.Concentration](#)

## Packages

- package [test](#)

## 7.22 /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/test/Print\_model.java File Reference

### Classes

- class [test.Print\\_model](#)

### Packages

- package [test](#)

## 7.23 /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/test/UnitCompareChainsRank.java File Reference

### Classes

- class [test.UnitCompareChainsRank](#)

### Packages

- package [test](#)

## 7.24 /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/test/UnitRankCompare.java File Reference

### Classes

- class [test.UnitRankCompare](#)

### Packages

- package [test](#)

## 7.25 /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/test/Utilities.java File Reference

## 7.26 /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/trie/Trie.java File Reference

### Classes

- class [trie.Trie](#)
- class [trie.Trie.TrieNode](#)

## Packages

- package [trie](#)

## 7.27 /home/element/PUF/Keyboard/java\_scripts/java\_marcov\_model/src/trie/TrieList.java File Reference

## Classes

- class [trie.TrieList](#)

## Packages

- package [trie](#)

# Index

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/components/Chain.java, 45

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/components/Distribution.↵  
java, 45

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/components/Token.java, 45

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/components/Touch.java, 46

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/components/Window.java, 46

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/data\_analysis/Model\_↵  
compare.java, 46

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/data\_analysis/Model\_↵  
compare\_thread.java, 47

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/data\_analysis/Statistics.↵  
java, 47

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/gui/Marcov\_console\_↵  
panel.java, 47

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/gui/Marcov\_file\_display\_↵  
\_panel.java, 47

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/gui/Marcov\_frame.java, 48

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/gui/Marcov\_options\_↵  
panel.java, 48

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/gui/StartGUI.java, 48

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/junit/Unit\_Compare\_↵  
ChainsRank.java, 48

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/junit/Unit\_Complete\_↵  
Probability.java, 49

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/rank/CompareChains\_↵  
Rank.java, 49

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/rank/CompleteProbability.↵  
java, 49

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/runtime/ChainBuilder.java, 49

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
\_marcov\_model/src/runtime/Compare\_↵  
Chains.java, 50

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/runtime/Operation\_↵  
thread.java, 50

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/test/Main.java, 50

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/test/Print\_model.java, 51

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
\_marcov\_model/src/test/UnitCompare\_↵  
ChainsRank.java, 51

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/test/UnitRankCompare.↵  
java, 51

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/test/Utilities.java, 51

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/trie/Trie.java, 51

/home/element/PUF/Keyboard/java\_scripts/java\_↵  
marcov\_model/src/trie/TrieList.java, 52

ABNORMAL

test::Main::TestFiles::Distribution, 21

add

trie::TrieList, 38

add\_touch

components::Chain, 12

add\_touch\_list

components::Chain, 12

addAll

trie::TrieList, 38, 39

auth\_chain

runtime::CompareChains, 17

authenticate

runtime::ChainBuilder, 15

authentication\_accuracy

data\_analysis::Statistics, 31

authentication\_probability

runtime::CompareChains, 17

average\_authentication\_probability

data\_analysis::Model\_compare\_thread, 28

best\_authentication\_percentage

data\_analysis::Statistics, 31

- build\_chain\_from\_csv
  - runtime::ChainBuilder, 15
- Chain
  - components::Chain, 12
- ChainBuilder
  - runtime::ChainBuilder, 14
- clear
  - trie::Trie, 37
  - trie::TrieList, 39
- close
  - gui::Marcov\_frame, 25
- combined
  - components::Token::Type, 40
- compare\_to
  - components::Chain, 12
- compare\_with\_token
  - components::Touch, 35
  - components::Window, 43
- CompareChains
  - runtime::CompareChains, 16
- CompareChainsRank
  - rank::CompareChainsRank, 18
- compareTo
  - components::Touch, 35
  - components::Window, 43
- complete
  - runtime::CompareChains, 17
- CompleteProbability
  - rank::CompleteProbability, 19
- components, 9
- components.Chain, 11
- components.Distribution, 22
- components.Token, 32
- components.Token.Type, 39
- components.Touch, 34
- components.Window, 42
- components::Chain
  - add\_touch, 12
  - add\_touch\_list, 12
  - Chain, 12
  - compare\_to, 12
  - compute\_uncomputed, 12
  - get\_distribution, 12
  - get\_key\_distribution, 12
  - get\_model\_size, 12
  - get\_threshold, 13
  - get\_token, 13
  - get\_tokens, 13
  - get\_touch\_probability, 13
  - get\_touches, 13
  - get\_window, 13
  - get\_windows, 13
  - is\_touch\_in\_key\_distribution, 13
  - output\_to\_csv, 13
  - reset, 13
  - set\_distribution, 13
  - toString, 13
- components::Distribution
  - Distribution, 22
  - equals, 22
  - get\_average, 22
  - get\_keycode, 23
  - get\_max, 23
  - get\_min, 23
  - get\_standard\_deviation, 23
  - update, 23
- components::Token
  - contains, 33
  - equals, 33
  - get\_acceptable\_wildcards, 33
  - get\_max, 33
  - get\_min, 33
  - get\_total\_wildcards, 33
  - increment\_high\_wildcards, 34
  - increment\_low\_wildcards, 34
  - is\_high\_wildcard, 34
  - is\_low\_wildcard, 34
  - Token, 32, 33
- components::Token::Type
  - combined, 40
  - keycode\_mu, 40
  - linear, 40
- components::Touch
  - compare\_with\_token, 35
  - compareTo, 35
  - get\_key, 35
  - get\_pressure, 36
  - get\_probability, 36
  - get\_timestamp, 36
  - hashCode, 36
  - set\_probability, 36
  - toString, 36
  - Touch, 35
- components::Window
  - compare\_with\_token, 43
  - compareTo, 43
  - get\_touch\_list, 43
  - hashCode, 43
  - size, 44
  - toString, 44
  - Window, 43
- compute\_probability
  - rank::CompleteProbability, 19
- compute\_uncomputed
  - components::Chain, 12
- Concentration
  - test::Main::TestFiles::Concentration, 20
- contains
  - components::Token, 33
- DISTRIBUTION
  - runtime::Operation\_thread::Computation, 19
- data\_analysis, 9
- data\_analysis.Model\_compare, 26
- data\_analysis.Model\_compare\_thread, 26
- data\_analysis.Statistics, 31
- data\_analysis::Model\_compare

- main, 26
- data\_analysis::Model\_compare\_thread
  - average\_authentication\_probability, 28
  - get\_auth\_data\_path, 27
  - get\_auth\_model\_size, 27
  - get\_auth\_probability\_list, 27
  - get\_base\_data\_path, 27
  - get\_base\_model\_size, 27
  - get\_threshold, 27
  - get\_token\_size, 27
  - get\_window\_size, 28
  - max\_authentication\_probability, 28
  - min\_authentication\_probability, 28
  - Model\_compare\_thread, 27
  - run, 28
- data\_analysis::Statistics
  - authentication\_accuracy, 31
  - best\_authentication\_percentage, 31
  - equal\_false\_positive\_negative\_authentication\_↔ percentage, 31
  - false\_negative\_percentage, 31
  - false\_positive\_percentage, 31
  - main, 31
  - minimize\_false\_positive\_authentication\_percentage, 31
- Distribution
  - components::Distribution, 22
  - test::Main::TestFiles::Distribution, 21
- equal\_false\_positive\_negative\_authentication\_↔ percentage
  - data\_analysis::Statistics, 31
- equals
  - components::Distribution, 22
  - components::Token, 33
- exit
  - gui::StartGUI, 30
- false\_negative\_percentage
  - data\_analysis::Statistics, 31
- false\_positive\_percentage
  - data\_analysis::Statistics, 31
- get\_acceptable\_wildcards
  - components::Token, 33
- get\_auth\_complete
  - runtime::CompareChains, 16
- get\_auth\_data\_path
  - data\_analysis::Model\_compare\_thread, 27
- get\_auth\_model\_size
  - data\_analysis::Model\_compare\_thread, 27
- get\_auth\_probability
  - runtime::CompareChains, 16
- get\_auth\_probability\_list
  - data\_analysis::Model\_compare\_thread, 27
- get\_auth\_result
  - runtime::CompareChains, 16
- get\_authenticate\_state
  - runtime::ChainBuilder, 15
- get\_authenticate\_thread
  - runtime::ChainBuilder, 15
- get\_average
  - components::Distribution, 22
- get\_base\_data\_path
  - data\_analysis::Model\_compare\_thread, 27
- get\_base\_model\_size
  - data\_analysis::Model\_compare\_thread, 27
- get\_distribution
  - components::Chain, 12
- get\_identifier
  - test::Main::TestFiles::Concentration, 20
  - test::Main::TestFiles::Distribution, 21
  - test::Main::TestFiles::PressureAmount, 29
- get\_index\_list
  - trie::Trie, 37
- get\_key
  - components::Touch, 35
- get\_key\_distribution
  - components::Chain, 12
- get\_keycode
  - components::Distribution, 23
- get\_max
  - components::Distribution, 23
  - components::Token, 33
- get\_min
  - components::Distribution, 23
  - components::Token, 33
- get\_model\_size
  - components::Chain, 12
- get\_pressure
  - components::Touch, 36
- get\_probability
  - components::Touch, 36
- get\_standard\_deviation
  - components::Distribution, 23
- get\_threshold
  - components::Chain, 13
  - data\_analysis::Model\_compare\_thread, 27
- get\_timestamp
  - components::Touch, 36
- get\_token
  - components::Chain, 13
- get\_token\_size
  - data\_analysis::Model\_compare\_thread, 27
- get\_tokens
  - components::Chain, 13
- get\_total\_wildcards
  - components::Token, 33
- get\_touch\_list
  - components::Window, 43
- get\_touch\_probability
  - components::Chain, 13
- get\_touches
  - components::Chain, 13
- get\_value
  - test::Main::TestFiles::Concentration, 20
  - test::Main::TestFiles::Distribution, 21

- test::Main::TestFiles::PressureAmount, 29
- get\_window
  - components::Chain, 13
- get\_window\_size
  - data\_analysis::Model\_compare\_thread, 28
- get\_windows
  - components::Chain, 13
- gui, 9
- gui.Marcov\_console\_panel, 24
- gui.Marcov\_file\_display\_panel, 24
- gui.Marcov\_frame, 25
- gui.Marcov\_options\_panel, 25
- gui.StartGUI, 30
- gui::Marcov\_console\_panel
  - Marcov\_console\_panel, 24
- gui::Marcov\_file\_display\_panel
  - Marcov\_file\_display\_panel, 24
- gui::Marcov\_frame
  - close, 25
  - Marcov\_frame, 25
- gui::Marcov\_options\_panel
  - Marcov\_options\_panel, 26
- gui::StartGUI
  - exit, 30
  - main, 30
- HIGH
  - test::Main::TestFiles::Concentration, 20
  - test::Main::TestFiles::PressureAmount, 29
- handle\_touch
  - runtime::ChainBuilder, 15
- hashCode
  - components::Touch, 36
  - components::Window, 43
- IN\_PROGRESS
  - runtime::ChainBuilder::State, 31
- increment\_high\_wildcards
  - components::Token, 34
- increment\_low\_wildcards
  - components::Token, 34
- init
  - junit::Unit\_CompareChainsRank, 40
  - junit::Unit\_CompleteProbability, 41
  - test::UnitCompareChainsRank, 41
  - test::UnitRankCompare, 42
- insertString
  - trie::Trie, 37
- is\_authentic
  - runtime::CompareChains, 17
- is\_high\_wildcard
  - components::Token, 34
- is\_low\_wildcard
  - components::Token, 34
- is\_touch\_in\_key\_distribution
  - components::Chain, 13
- junit, 10
- junit.Unit\_CompareChainsRank, 40
- junit.Unit\_CompleteProbability, 40
- junit::Unit\_CompareChainsRank
  - init, 40
  - test\_authentication\_probability, 40
- junit::Unit\_CompleteProbability
  - init, 41
  - test\_replica\_distribution, 41
- KEY\_DISTRIBUTION
  - runtime::Operation\_thread::Computation, 19
- keycode\_mu
  - components::Token::Type, 40
- LOW
  - test::Main::TestFiles::Concentration, 20
  - test::Main::TestFiles::PressureAmount, 29
- linear
  - components::Token::Type, 40
- MEDIUM
  - test::Main::TestFiles::Concentration, 20
  - test::Main::TestFiles::PressureAmount, 29
- main
  - data\_analysis::Model\_compare, 26
  - data\_analysis::Statistics, 31
  - gui::StartGUI, 30
  - test::Main, 23
  - test::Print\_model, 30
- Marcov\_console\_panel
  - gui::Marcov\_console\_panel, 24
- Marcov\_file\_display\_panel
  - gui::Marcov\_file\_display\_panel, 24
- Marcov\_frame
  - gui::Marcov\_frame, 25
- Marcov\_options\_panel
  - gui::Marcov\_options\_panel, 26
- max\_authentication\_probability
  - data\_analysis::Model\_compare\_thread, 28
- min\_authentication\_probability
  - data\_analysis::Model\_compare\_thread, 28
- minimize\_false\_positive\_authentication\_percentage
  - data\_analysis::Statistics, 31
- Model\_compare\_thread
  - data\_analysis::Model\_compare\_thread, 27
- NORMAL
  - test::Main::TestFiles::Distribution, 21
- occurrence\_count
  - trie::Trie, 37
  - trie::TrieList, 39
- Operation\_thread
  - runtime::Operation\_thread, 28
- output\_to\_csv
  - components::Chain, 13
- PROBABILITY\_VECTOR\_DIFFERENCE
  - runtime::ChainBuilder::CompareMethod, 18
- PROBABILITY
  - runtime::Operation\_thread::Computation, 19



- parse\_csv
  - runtime::ChainBuilder, 15
- PressureAmount
  - test::Main::TestFiles::PressureAmount, 29
- printSorted
  - trie::Trie, 37
- RANDOM
  - test::Main::TestFiles::Distribution, 21
- rank, 10
- rank.CompareChainsRank, 17
- rank.CompleteProbability, 18
- rank::CompareChainsRank
  - CompareChainsRank, 18
  - run, 18
- rank::CompleteProbability
  - CompleteProbability, 19
  - compute\_probability, 19
- remove
  - trie::TrieList, 39
- removeAll
  - trie::TrieList, 39
- reset
  - components::Chain, 13
- retainAll
  - trie::TrieList, 39
- run
  - data\_analysis::Model\_compare\_thread, 28
  - rank::CompareChainsRank, 18
  - runtime::CompareChains, 16
  - runtime::Operation\_thread, 28
- runtime, 10
- runtime.ChainBuilder, 14
- runtime.ChainBuilder.CompareMethod, 18
- runtime.ChainBuilder.State, 30
- runtime.CompareChains, 15
- runtime.Operation\_thread, 28
- runtime.Operation\_thread.Computation, 19
- runtime::ChainBuilder
  - authenticate, 15
  - build\_chain\_from\_csv, 15
  - ChainBuilder, 14
  - get\_authenticate\_state, 15
  - get\_authenticate\_thread, 15
  - handle\_touch, 15
  - parse\_csv, 15
- runtime::ChainBuilder::CompareMethod
  - PROBABILITY\_VECTOR\_DIFFERENCE, 18
- runtime::ChainBuilder::State
  - IN\_PROGRESS, 31
  - SUCCESS, 31
- runtime::CompareChains
  - auth\_chain, 17
  - authentication\_probability, 17
  - CompareChains, 16
  - complete, 17
  - get\_auth\_complete, 16
  - get\_auth\_probability, 16
  - get\_auth\_result, 16
  - is\_authentic, 17
  - run, 16
  - user\_chain, 17
- runtime::Operation\_thread
  - Operation\_thread, 28
  - run, 28
- runtime::Operation\_thread::Computation
  - DISTRIBUTION, 19
  - KEY\_DISTRIBUTION, 19
  - PROBABILITY, 19
  - TOKEN, 19
  - WINDOW, 19
- SUCCESS
  - runtime::ChainBuilder::State, 31
- set
  - trie::TrieList, 39
- set\_distribution
  - components::Chain, 13
- set\_probability
  - components::Touch, 36
- set\_tokens
  - trie::TrieList, 39
- size
  - components::Window, 44
- successor\_count
  - trie::TrieList, 39
- TOKEN
  - runtime::Operation\_thread::Computation, 19
- test, 10
  - test::UnitCompareChainsRank, 41
  - test::UnitRankCompare, 42
- test.Main, 23
- test.Main.TestFiles.Concentration, 20
- test.Main.TestFiles.Distribution, 21
- test.Main.TestFiles.PressureAmount, 29
- test.Print\_model, 30
- test.UnitCompareChainsRank, 41
- test.UnitRankCompare, 42
- test::Main
  - main, 23
- test::Main::TestFiles::Concentration
  - Concentration, 20
  - get\_identifier, 20
  - get\_value, 20
  - HIGH, 20
  - LOW, 20
  - MEDIUM, 20
  - toString, 20
- test::Main::TestFiles::Distribution
  - ABNORMAL, 21
  - Distribution, 21
  - get\_identifier, 21
  - get\_value, 21
  - NORMAL, 21
  - RANDOM, 21
  - toString, 21
- test::Main::TestFiles::PressureAmount

- get\_identifier, [29](#)
  - get\_value, [29](#)
  - HIGH, [29](#)
  - LOW, [29](#)
  - MEDIUM, [29](#)
  - PressureAmount, [29](#)
  - toString, [29](#)
- test::Print\_model
  - main, [30](#)
- test::UnitCompareChainsRank
  - init, [41](#)
  - test, [41](#)
  - test\_chain\_to\_graph, [41](#)
  - test\_touch\_index, [41](#)
  - test\_touch\_window, [41](#)
- test::UnitRankCompare
  - init, [42](#)
  - test, [42](#)
  - test\_auth\_probability, [42](#)
  - test\_compare\_correct, [42](#)
- test\_auth\_probability
  - test::UnitRankCompare, [42](#)
- test\_authentication\_probability
  - junit::Unit\_CompareChainsRank, [40](#)
- test\_chain\_to\_graph
  - test::UnitCompareChainsRank, [41](#)
- test\_compare\_correct
  - test::UnitRankCompare, [42](#)
- test\_replica\_distribution
  - junit::Unit\_CompleteProbability, [41](#)
- test\_touch\_index
  - test::UnitCompareChainsRank, [41](#)
- test\_touch\_window
  - test::UnitCompareChainsRank, [41](#)
- toString
  - components::Chain, [13](#)
  - components::Touch, [36](#)
  - components::Window, [44](#)
  - test::Main::TestFiles::Concentration, [20](#)
  - test::Main::TestFiles::Distribution, [21](#)
  - test::Main::TestFiles::PressureAmount, [29](#)
- Token
  - components::Token, [32](#), [33](#)
- Touch
  - components::Touch, [35](#)
- Trie
  - trie::Trie, [37](#)
- trie, [10](#)
- trie.Trie, [36](#)
- trie.TrieList, [38](#)
- trie::Trie
  - clear, [37](#)
  - get\_index\_list, [37](#)
  - insertString, [37](#)
  - occurrence\_count, [37](#)
  - printSorted, [37](#)
  - Trie, [37](#)
- trie::TrieList
  - add, [38](#)
  - addAll, [38](#), [39](#)
  - clear, [39](#)
  - occurrence\_count, [39](#)
  - remove, [39](#)
  - removeAll, [39](#)
  - retainAll, [39](#)
  - set, [39](#)
  - set\_tokens, [39](#)
  - successor\_count, [39](#)
  - TrieList, [38](#)
- TrieList
  - trie::TrieList, [38](#)
- update
  - components::Distribution, [23](#)
- user\_chain
  - runtime::CompareChains, [17](#)
- WINDOW
  - runtime::Operation\_thread::Computation, [19](#)
- Window
  - components::Window, [43](#)