

Bayesian Cluster Tool

Generated by Doxygen 1.9.1

Thu Jul 13 2023 18:15:41

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 ClusterWrapper Struct Reference	5
3.1.1 Detailed Description	5
3.1.2 Member Function Documentation	6
3.1.2.1 operator<()	6
3.1.2.2 operator==()	6
3.2 ScanEntry Struct Reference	7
3.2.1 Detailed Description	7
3.2.2 Member Function Documentation	7
3.2.2.1 operator<()	7
3.2.2.2 operator==()	8
4 File Documentation	9
4.1 include/BayesianClustering/API.hpp File Reference	9
4.1.1 Function Documentation	11
4.1.1.1 _ClusterCallback_Json_()	11
4.1.1.2 _FullClusterToSimpleCluster_()	11
4.1.1.3 _FullScanToSimpleScan_()	11
4.1.1.4 _ScanCallback_Json_()	12
4.1.1.5 RunClustering() [1/3]	12
4.1.1.6 RunClustering() [2/3]	13
4.1.1.7 RunClustering() [3/3]	13
4.1.1.8 RunScan() [1/3]	14
4.1.1.9 RunScan() [2/3]	14
4.1.1.10 RunScan() [3/3]	15
4.2 src/BayesianClustering/API.cpp File Reference	15
4.2.1 Function Documentation	16
4.2.1.1 _ClusterCallback_Json_()	16
4.2.1.2 _FullClusterToSimpleCluster_()	17
4.2.1.3 _FullScanToSimpleScan_()	17
4.2.1.4 _ScanCallback_Json_()	17
Index	19

Chapter 1

Class Index

1.1 Class List

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

ClusterWrapper	A struct for storing extracted parameters from a cluster	5
ScanEntry	A struct for storing a result of an individual scan configuration	7

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

include/BayesianClustering/ API.hpp	9
src/BayesianClustering/ API.cpp	15

Chapter 3

Class Documentation

3.1 ClusterWrapper Struct Reference

A struct for storing extracted parameters from a cluster.

```
#include <API.hpp>
```

Public Member Functions

- bool `operator<` (const `ClusterWrapper` &aOther)
Comparison operator for sorting.
- bool `operator==` (const `ClusterWrapper` &aOther)
Equality operator required by boost python.

Public Attributes

- `std::size_t` `localizations`
The number of localizations in the cluster.
- `long double` `area`
The area of the spanning convex hull.
- `long double` `perimeter`
The perimeter of the spanning convex hull.
- `double` `centroid_x`
The x-position of the centroid.
- `double` `centroid_y`
The y-position of the centroid.

3.1.1 Detailed Description

A struct for storing extracted parameters from a cluster.

Definition at line 47 of file API.hpp.

3.1.2 Member Function Documentation

3.1.2.1 `operator<()`

```
bool ClusterWrapper::operator< (
    const ClusterWrapper & aOther ) [inline]
```

Comparison operator for sorting.

Returns

Whether we are smaller than the other

Parameters

<i>aOther</i>	Another ClusterWrapper to compare against
---------------	---

Definition at line 57 of file API.hpp.

References area, localizations, and perimeter.

3.1.2.2 `operator==()`

```
bool ClusterWrapper::operator== (
    const ClusterWrapper & aOther ) [inline]
```

Equality operator required by boost python.

Returns

Whether we are equal to the other

Parameters

<i>aOther</i>	Another ClusterWrapper to compare against
---------------	---

Definition at line 67 of file API.hpp.

References centroid_x, and centroid_y.

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

- include/BayesianClustering/[API.hpp](#)

3.2 ScanEntry Struct Reference

A struct for storing a result of an individual scan configuration.

```
#include <API.hpp>
```

Public Member Functions

- bool `operator<` (const `ScanEntry` &aOther)
Comparison operator for sorting.
- bool `operator==` (const `ScanEntry` &aOther)
Equality operator required by boost python.

Public Attributes

- double `r`
The R parameter
- double `t`
The T parameter.
- double `score`
The score.

3.2.1 Detailed Description

A struct for storing a result of an individual scan configuration.

Definition at line 19 of file API.hpp.

3.2.2 Member Function Documentation

3.2.2.1 `operator<()`

```
bool ScanEntry::operator< (  
    const ScanEntry & aOther ) [inline]
```

Comparison operator for sorting.

Returns

Whether we are smaller than the other

Parameters

<i>aOther</i>	Another ScanEntry to compare against
---------------	--

Definition at line 27 of file API.hpp.

References [r](#), and [t](#).

3.2.2.2 operator==()

```
bool ScanEntry::operator== (
    const ScanEntry & aOther ) [inline]
```

Equality operator required by boost python.

Returns

Whether we are equal to the other

Parameters

<i>aOther</i>	Another ScanEntry to compare against
---------------	--

Definition at line 36 of file API.hpp.

References [r](#), [score](#), and [t](#).

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

- [include/BayesianClustering/API.hpp](#)

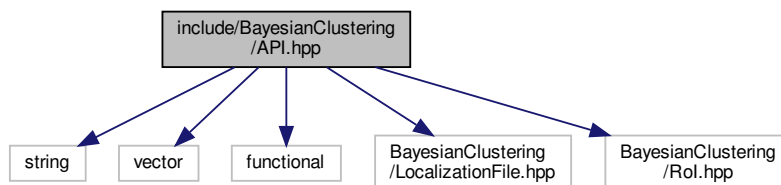
Chapter 4

File Documentation

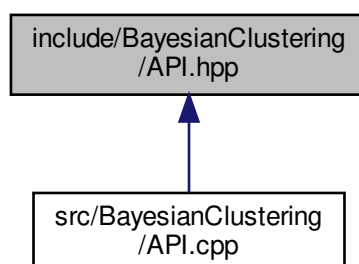
4.1 include/BayesianClustering/API.hpp File Reference

```
#include <string>
#include <vector>
#include <functional>
#include "BayesianClustering/LocalizationFile.hpp"
#include "BayesianClustering/RoI.hpp"
```

Include dependency graph for API.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct [ScanEntry](#)
A struct for storing a result of an individual scan configuration.
- struct [ClusterWrapper](#)
A struct for storing extracted parameters from a cluster.

Typedefs

- typedef std::function< void(Rolproxy &, const double &, const double &) > [tFullScanCallback](#)
Typedef the full scan callback for clarity.
- typedef std::function< void(const std::string &, const std::vector< [ScanEntry](#) > &) > [tSimpleScanCallback](#)
Typedef the simplified scan callback for clarity.
- typedef std::function< void(Rolproxy &) > [tFullClusterCallback](#)
Typedef the full clustering callback for clarity.
- typedef std::function< void(const std::string &, const std::vector< [ClusterWrapper](#) > &) > [tSimpleClusterCallback](#)
Typedef the simplified clustering callback for clarity.

Functions

- void [_ScanCallback_Json_](#) (const std::string &aRoild, const std::vector< [ScanEntry](#) > &aVector, const std::string &aInFile, const std::string &aOutputPattern)
A callback to dump a scan to a JSON file.
- void [_FullScanToSimpleScan_](#) (Rol &aRol, const ScanConfiguration &aScanConfig, const [tSimpleScanCallback](#) &aCallback)
A callback to neatly package the scan results for easy consumption.
- void [_ClusterCallback_Json_](#) (const std::string &aRoild, const std::vector< [ClusterWrapper](#) > &aVector, const std::string &aInFile, const std::string &aOutputPattern)
A callback to dump a clustering run to a JSON file.
- void [_FullClusterToSimpleCluster_](#) (Rolproxy &aRolproxy, const [tSimpleClusterCallback](#) &aCallback)
A callback to neatly package the scan results for easy consumption.
- template<typename RolConfig >
void [RunScan](#) (const std::string &aInFile, const RolConfig &aRolConfig, const ScanConfiguration &aScanConfig, const [tFullScanCallback](#) &aCallback)
Automatically extract Rol, run scan and apply a full call-back.
- template<typename RolConfig >
void [RunScan](#) (const std::string &aInFile, const RolConfig &aRolConfig, const ScanConfiguration &aScanConfig, const [tSimpleScanCallback](#) &aCallback)
Automatically extract Rol, run scan and apply a simple call-back.
- template<typename RolConfig >
void [RunScan](#) (const std::string &aInFile, const RolConfig &aRolConfig, const ScanConfiguration &aScanConfig, const std::string &aOutputPattern)
Automatically extract Rol, run scan and dump to JSON file.
- template<typename RolConfig >
void [RunClustering](#) (const std::string &aInFile, const RolConfig &aRolConfig, const double &aR, const double &aT, const [tFullClusterCallback](#) &aCallback)
Automatically extract Rol, clusterize and apply a full call-back.
- template<typename RolConfig >
void [RunClustering](#) (const std::string &aInFile, const RolConfig &aRolConfig, const double &aR, const double &aT, const [tSimpleClusterCallback](#) &aCallback)
Automatically extract Rol, clusterize and apply a full call-back.
- template<typename RolConfig >
void [RunClustering](#) (const std::string &aInFile, const RolConfig &aRolConfig, const double &aR, const double &aT, const std::string &aOutputPattern)
Automatically specify Rol, clusterize and apply a full call-back.

4.1.1 Function Documentation

4.1.1.1 _ClusterCallback_Json_()

```
void _ClusterCallback_Json_ (
    const std::string & aRoId,
    const std::vector< ClusterWrapper > & aVector,
    const std::string & aInFile,
    const std::string & aOutputPattern )
```

A callback to dump a clustering run to a JSON file.

Parameters

<i>aRoId</i>	The RoI ID
<i>aVector</i>	A vector of cluster-wrappers
<i>aInFile</i>	The name of the localization file
<i>aOutputPattern</i>	The name of the output JSON file

Definition at line 87 of file API.cpp.

Referenced by RunClustering().

4.1.1.2 _FullClusterToSimpleCluster_()

```
void _FullClusterToSimpleCluster_ (
    RoIproxy & aRoIproxy,
    const tSimpleClusterCallback & aCallback )
```

A callback to neatly package the scan results for easy consumption.

Parameters

<i>aRoIproxy</i>	The region-proxy containing the clusters
<i>aCallback</i>	The simple callback to be applied

Definition at line 106 of file API.cpp.

Referenced by RunClustering().

4.1.1.3 _FullScanToSimpleScan_()

```
void _FullScanToSimpleScan_ (
    RoI & aRoI,
```

```
const ScanConfiguration & aScanConfig,
const tSimpleScanCallback & aCallback )
```

A callback to neatly package the scan results for easy consumption.

Parameters

<i>aRol</i>	The region of interest
<i>aScanConfig</i>	The configuration for the scan
<i>aCallback</i>	The simple callback to be applied

Definition at line 72 of file API.cpp.

Referenced by RunScan().

4.1.1.4 _ScanCallback_Json_()

```
void _ScanCallback_Json_ (
    const std::string & aRoiId,
    const std::vector< ScanEntry > & aVector,
    const std::string & aInFile,
    const std::string & aOutputPattern )
```

A callback to dump a scan to a JSON file.

Parameters

<i>aRoiId</i>	The RoI ID
<i>aVector</i>	A vector of scan results
<i>aInFile</i>	The name of the localization file
<i>aOutputPattern</i>	The name of the output JSON file

Definition at line 26 of file API.cpp.

Referenced by RunScan().

4.1.1.5 RunClustering() [1/3]

```
template<typename RoIConfig >
void RunClustering (
    const std::string & aInFile,
    const RoIConfig & aRoIConfig,
    const double & aR,
    const double & aT,
    const std::string & aOutputPattern )
```

Automatically specify RoI, clusterize and apply a full call-back.

Parameters

<i>aInFile</i>	The name of the localization file
<i>aRoIConfig</i>	Specify the mechanism used to extract the Rols
<i>aR</i>	The R value of the clusterizer
<i>aT</i>	The T value of the clusterizer
<i>aOutputPattern</i>	A formattable-string specifying the name of the output JSON files. Substitutable fields are {input} (giving the stem of the input file name) and {roi} (giving the RoI id).

Definition at line 181 of file API.hpp.

References `_ClusterCallback_Json_()`, and `RunClustering()`.

4.1.1.6 RunClustering() [2/3]

```
template<typename RoIConfig >
void RunClustering (
    const std::string & aInFile,
    const RoIConfig & aRoIConfig,
    const double & aR,
    const double & aT,
    const tFullClusterCallback & aCallback )
```

Automatically extract RoI, clusterize and apply a full call-back.

Parameters

<i>aInFile</i>	The name of the localization file
<i>aRoIConfig</i>	Specify the mechanism used to extract the Rols
<i>aR</i>	The R value of the clusterizer
<i>aT</i>	The T value of the clusterizer
<i>aCallback</i>	The callback to be applied

Definition at line 157 of file API.hpp.

Referenced by `RunClustering()`.

4.1.1.7 RunClustering() [3/3]

```
template<typename RoIConfig >
void RunClustering (
    const std::string & aInFile,
    const RoIConfig & aRoIConfig,
    const double & aR,
    const double & aT,
    const tSimpleClusterCallback & aCallback )
```

Automatically extract RoI, clusterize and apply a full call-back.

Parameters

<i>aInFile</i>	The name of the localization file
<i>aRoIConfig</i>	Specify the mechanism used to extract the Rols
<i>aR</i>	The R value of the clusterizer
<i>aT</i>	The T value of the clusterizer
<i>aCallback</i>	The callback to be applied

Definition at line 169 of file API.hpp.

References `_FullClusterToSimpleCluster_()`, and `RunClustering()`.

4.1.1.8 RunScan() [1/3]

```
template<typename RoIConfig >
void RunScan (
    const std::string & aInFile,
    const RoIConfig & aRoIConfig,
    const ScanConfiguration & aScanConfig,
    const std::string & aOutputPattern )
```

Automatically extract RoI, run scan and dump to JSON file.

Parameters

<i>aInFile</i>	The name of the localization file
<i>aRoIConfig</i>	Specify the mechanism used to extract the Rols
<i>aScanConfig</i>	The configuration for the scan
<i>aOutputPattern</i>	A formattable-string specifying the name of the output JSON files. Substitutable fields are {input} (giving the stem of the input file name) and {roi} (giving the RoI id).

Definition at line 142 of file API.hpp.

References `_ScanCallback_Json_()`, and `RunScan()`.

4.1.1.9 RunScan() [2/3]

```
template<typename RoIConfig >
void RunScan (
    const std::string & aInFile,
    const RoIConfig & aRoIConfig,
    const ScanConfiguration & aScanConfig,
    const tFullScanCallback & aCallback )
```

Automatically extract RoI, run scan and apply a full call-back.

Parameters

<i>aInFile</i>	The name of the localization file
<i>aRoIConfig</i>	Specify the mechanism used to extract the Rols
<i>aScanConfig</i>	The configuration for the scan
<i>aCallback</i>	The full callback to be applied

Definition at line 120 of file API.hpp.

Referenced by RunScan().

4.1.1.10 RunScan() [3/3]

```
template<typename RoIConfig >
void RunScan (
    const std::string & aInFile,
    const RoIConfig & aRoIConfig,
    const ScanConfiguration & aScanConfig,
    const tSimpleScanCallback & aCallback )
```

Automatically extract RoI, run scan and apply a simple call-back.

Parameters

<i>aInFile</i>	The name of the localization file
<i>aRoIConfig</i>	Specify the mechanism used to extract the Rols
<i>aScanConfig</i>	The configuration for the scan
<i>aCallback</i>	The simple callback to be applied

Definition at line 131 of file API.hpp.

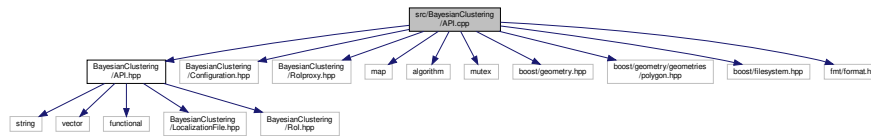
References _FullScanToSimpleScan_().

4.2 src/BayesianClustering/API.cpp File Reference

```
#include "BayesianClustering/API.hpp"
#include "BayesianClustering/Configuration.hpp"
#include "BayesianClustering/RoIproxy.hpp"
#include <map>
#include <algorithm>
#include <mutex>
#include <boost/geometry.hpp>
#include <boost/geometry/geometries/polygon.hpp>
#include <boost/filesystem.hpp>
```

```
#include <fmt/format.h>
```

Include dependency graph for API.cpp:



Functions

- void [_ScanCallback_Json_](#) (const std::string &aRoild, const std::vector< [ScanEntry](#) > &aVector, const std::string &alnFile, const std::string &aOutputPattern)
A callback to dump a scan to a JSON file.
- void [_FullScanToSimpleScan_](#) (Rol &aRol, const ScanConfiguration &aScanConfig, const [tSimpleScanCallback](#) &aCallback)
A callback to neatly package the scan results for easy consumption.
- void [_ClusterCallback_Json_](#) (const std::string &aRoild, const std::vector< [ClusterWrapper](#) > &aVector, const std::string &alnFile, const std::string &aOutputPattern)
A callback to dump a clustering run to a JSON file.
- void [_FullClusterToSimpleCluster_](#) (Rolproxy &aRolproxy, const [tSimpleClusterCallback](#) &aCallback)
A callback to neatly package the scan results for easy consumption.

4.2.1 Function Documentation

4.2.1.1 [_ClusterCallback_Json_\(\)](#)

```
void _ClusterCallback_Json_ (
    const std::string & aRoild,
    const std::vector< ClusterWrapper > & aVector,
    const std::string & aInFile,
    const std::string & aOutputPattern )
```

A callback to dump a clustering run to a JSON file.

Parameters

<i>aRoild</i>	The Rol ID
<i>aVector</i>	A vector of cluster-wrappers
<i>alnFile</i>	The name of the localization file
<i>aOutputPattern</i>	The name of the output JSON file

Definition at line 87 of file API.cpp.

Referenced by [RunClustering\(\)](#).

4.2.1.2 _FullClusterToSimpleCluster_()

```
void _FullClusterToSimpleCluster_ (
    RoIproxy & aRoIproxy,
    const tSimpleClusterCallback & aCallback )
```

A callback to neatly package the scan results for easy consumption.

Parameters

<i>aRoIproxy</i>	The region-proxy containing the clusters
<i>aCallback</i>	The simple callback to be applied

Definition at line 106 of file API.cpp.

Referenced by RunClustering().

4.2.1.3 _FullScanToSimpleScan_()

```
void _FullScanToSimpleScan_ (
    RoI & aRoI,
    const ScanConfiguration & aScanConfig,
    const tSimpleScanCallback & aCallback )
```

A callback to neatly package the scan results for easy consumption.

Parameters

<i>aRoI</i>	The region of interest
<i>aScanConfig</i>	The configuration for the scan
<i>aCallback</i>	The simple callback to be applied

Definition at line 72 of file API.cpp.

Referenced by RunScan().

4.2.1.4 _ScanCallback_Json_()

```
void _ScanCallback_Json_ (
    const std::string & aRoId,
    const std::vector< ScanEntry > & aVector,
    const std::string & aInFile,
    const std::string & aOutputPattern )
```

A callback to dump a scan to a JSON file.

Parameters

<i>aRoId</i>	The RoI ID
<i>aVector</i>	A vector of scan results
<i>aInFile</i>	The name of the localization file
<i>aOutputPattern</i>	The name of the output JSON file

Definition at line 26 of file API.cpp.

Referenced by RunScan().

Index

- `_ClusterCallback_Json_`
 - API.cpp, [16](#)
 - API.hpp, [11](#)
 - `_FullClusterToSimpleCluster_`
 - API.cpp, [16](#)
 - API.hpp, [11](#)
 - `_FullScanToSimpleScan_`
 - API.cpp, [17](#)
 - API.hpp, [11](#)
 - `_ScanCallback_Json_`
 - API.cpp, [17](#)
 - API.hpp, [12](#)
- API.cpp
 - `_ClusterCallback_Json_`, [16](#)
 - `_FullClusterToSimpleCluster_`, [16](#)
 - `_FullScanToSimpleScan_`, [17](#)
 - `_ScanCallback_Json_`, [17](#)
- API.hpp
 - `_ClusterCallback_Json_`, [11](#)
 - `_FullClusterToSimpleCluster_`, [11](#)
 - `_FullScanToSimpleScan_`, [11](#)
 - `_ScanCallback_Json_`, [12](#)
 - RunClustering, [12](#), [13](#)
 - RunScan, [14](#), [15](#)
- ClusterWrapper, [5](#)
 - operator<, [6](#)
 - operator==, [6](#)
- include/BayesianClustering/API.hpp, [9](#)
- operator<
 - ClusterWrapper, [6](#)
 - ScanEntry, [7](#)
- operator==
 - ClusterWrapper, [6](#)
 - ScanEntry, [8](#)
- RunClustering
 - API.hpp, [12](#), [13](#)
- RunScan
 - API.hpp, [14](#), [15](#)
- ScanEntry, [7](#)
 - operator<, [7](#)
 - operator==, [8](#)
- src/BayesianClustering/API.cpp, [15](#)