
Basic Applied Topology Subprograms

Brad Nelson

Jul 12, 2022

CONTENTS:

1	Why BATS?	3
1.1	Installation	3
1.2	Quick Start Guide	4
1.3	Tutorials	10
1.4	Examples	10
1.5	Library API	12
1.6	About BATS	246
2	Indices and tables	247
	Index	249

The Basic Applied Topology Subprograms (BATS) are a C++ header-only library for applied algebraic topology. It includes functionality for

- Creation of simplicial, cubical, and cell complexes, as well as cellular maps
- Implementation of chain and homology functors
- Filtered complexes and persistent homology
- Diagrams of spaces and maps, and computation of zigzag homology
- Topological constructions such as Vietoris-Rips complexes, Witness complexes, Nerves of covers
- Sparse linear algebra over (finite) fields
- And more!

BATS attempts to balance performance with extensibility. The library uses template metaprogramming to make it easy to swap in and out different constructions while maintaining a consistent core functionality.

WHY BATS?

There are many very high-performance libraries for computing things like persistent homology that have been developed over the past decade. Unlike many of these libraries BATS is focused on functoriality, and provides functionality to handle maps between topological spaces, chain maps, and induced maps on homology. The goal is to make it easier for researchers and practitioners to implement and explore the vast back catalog of algebraic topology while also providing applied functionality.

1.1 Installation

You can obtain BATS from GitHub:

```
$ git clone https://github.com/bnel/BATS.git
```

1.1.1 Dependencies

BATS is written in C++ and requires a compiler which is able to understand C++17 syntax. It has been run and tested on Linux (Ubuntu and Fedora distributions) using the gcc compiler.

BATS itself is header-only, and is built using STL data structures, meaning you don't need to find other headers.

To get the most out of BATS, you should compile with OpenMP.

```
$ g++ ...<other flags> -std=c++17 -fopenmp
```

1.1.2 Testing Installation

Once you have BATS on your computer, you can test your installation. from the root of your repository:

```
$ cd tests  
$ make test -j
```

1.2 Quick Start Guide

This guide will cover several basic use cases for BATS. More specialized functionality is covered in the tutorials and examples. Ultimately, you can use the API reference.

1.2.1 Simplicial Complexes and Homology

BATS offers two implementations of simplicial complexes: `SimplicialComplex` and `LightSimplicialComplex`. While the internal representations differ, they both have the same interface which can be used. When dealing with stand-alone simplices, BATS uses `std::vector<size_t>` to represent simplices, from which the vertex set and dimension of the simplex can be extracted.

Warning: Simplices in bats should generally be assumed to be *ordered*, meaning that $\{0, 1, 2\}$ is not the same as $\{1, 2, 0\}$. If you want to use *unordered* simplices, you can either add vertices in sorted order, or use a sorting algorithm before adding simplices to complexes.

The two main methods for adding simplices to simplicial complexes are `add`, which assumes you have already added the boundary of a simplex, and `add_recursive` which will add any faces that are not already present.

```
bats::SimplicialComplex X();
X.add_recursive({0,1,2});
X.add_recursive({2,3});
X.add({1,3});

X.print_summary();
```

The above code will create a `SimplicialComplex` with a single connected component and a single hole. The call to `X.print_summary()` will produce

```
SimplicialComplex, maxdim = 2
  dim 0 : 4 cells
  dim 1 : 5 cells
  dim 2 : 1 cells
10 total
```

Let's now compute homology.

```
using F2 = ModP<int, 2>;
auto R = bats::Reduce(X, F2());

R.print_summary();
```

The output of `bats::Reduce` will be a `ReducedChainComplex` with F2 coefficients, which holds information used to compute homology. The output of `R.print_summary()` will be

```
ReducedChainComplex, maxdim = 2
  dim 0: 4, betti_0: 1
  dim 1: 5, betti_1: 1
  dim 2: 1, betti_2: 0
```


1.2.2 Persistent Homology

A filtration in BATS is a class which is templated over the type of the filtration values as well as the type of the underlying complex.

```
bats::Filtration<double, bats::SimplicialComplex> F;
std::vector<size_t> spx;
spx = {0,1,2}; F.add_recursive(0.0, spx);
spx = {2,3}; F.add_recursive(1.0, spx);
spx = {1,3}; F.add(2.0, spx);

F.complex().print_summary();
```

The underlying `SimplicialComplex` is the same as in the previous example:

```
SimplicialComplex, maxdim = 2
  dim 0 : 4 cells
  dim 1 : 5 cells
  dim 2 : 1 cells
10 total
```

Again, we can use the `Reduce` function to compute homology. Because we are using a filtration as input, persistent homology will be computed, returning a `ReducedFilteredChainComplex`.

```
using F2 = ModP<int, 2>;
auto R = bats::Reduce(F, F2());

for (size_t k = 0; k < R.maxdim(); ++k) {
  std::cout << "\n" << k << "-dimensional barcode:\n";
  for (auto p : R.persistence_pairs(k)) {
    std::cout << p.str() << std::endl;
  }
}
```

The output will show one persistent 0-dimensional homology class as well as one persistent 1-dimensional homology class

```
0-dimensional barcode:
0 : (0,inf) <0,-1>
0 : (0,0) <1,0>
0 : (0,0) <2,1>
0 : (1,1) <3,3>

1-dimensional barcode:
1 : (0,0) <2,0>
1 : (2,inf) <4,-1>
```

The output of `R.persistence_pairs(k)` is a vector of `PersistencePairs` for k -dimensional persistent homology.

A `PersistencePair` includes 5 pieces of information: * The dimension of the homology class. * The birth and death parameters of the homology class. * The simplex indices responsible for birth and death.

1.2.3 Maps

BATS makes dealing with maps between topological spaces and associated chain maps and induced maps on homology easy. The relevant class is a `CellularMap` which keeps track of what cells in one complex map to what cells in another.

We'll just look at a wrapper for `CellularMap`, called `SimplicialMap` which can be used to extend a map on the vertex set of a `SimplicialComplex` to a map of simplices.

First, we'll build identical simplicial complexes `X` and `Y` which are both cycle graphs on four vertices.

```
bats::SimplicialComplex X;
X.add_recursive({0,1});
X.add_recursive({1,2});
X.add_recursive({2,3});
X.add_recursive({0,3});
bats::SimplicialComplex Y = X; // copy
```

We then build a simplicial map from `X` to `Y` which is extended from a reflection of the vertices.

```
std::vector<size_t> f0{2,1,0,3}; // reflection map
auto F = bats::SimplicialMap(X, Y, f0);
```

The map is extended by sending vertex `i` in `X` to vertex `f0[i]` in `Y`. Next, we can apply the chain functor. We'll use `F3` coefficients.

```
// apply the chain functor
using F3 = ModP<int, 3>;
auto CX = bats::Chain(X, F3());
auto CY = bats::Chain(Y, F3());
auto CF = bats::Chain(F, F3());
```

Finally, we can compute homology of the chain complexes and the induced maps.

```
auto RX = bats::ReducedChainComplex(CX);
auto RY = bats::ReducedChainComplex(CY);
RX.print_summary();
RY.print_summary();

for (size_t k = 0; k < 2; k++) {
    std::cout << "\nInduced map in dimension " << k << std::endl;
    auto HF = bats::induced_map(CF, RX, RY, k);
    HF.print();
}
```

The following output will be produced:

```
ReducedChainComplex, maxdim = 1
  dim 0: 4, betti_0: 1
  dim 1: 4, betti_1: 1
ReducedChainComplex, maxdim = 1
  dim 0: 4, betti_0: 1
  dim 1: 4, betti_1: 1

Induced map in dimension 0
[0x7fff6f336460] : 1 x 1 ColumnMatrix
```

(continues on next page)

(continued from previous page)

```

1
Induced map in dimension 1
[0x7fff6f336460] : 1 x 1 ColumnMatrix
2

```

As expected, we see that X and Y both have 1-dimensional homology in dimensions 0 and 1. The induced map in dimension 0 is the identity, and the induced map in dimension 1 is multiplication by -1 (in mod-3 coefficients).

1.2.4 Zigzag Homology

We'll now compute a simple zigzag barcode, using the above example. We'll consider a diagram with two (identical) spaces, connected by a single edge which applies the reflection map in the above example.

```

bats::Diagram<bats::SimplicialComplex, bats::CellularMap> D(2,1);

bats::SimplicialComplex X;
X.add_recursive({0,1});
X.add_recursive({1,2});
X.add_recursive({2,3});
X.add_recursive({0,3});

std::vector<size_t> f0{2,1,0,3}; // reflection map
auto F = bats::SimplicialMap(X, X, f0);

D.set_node(0, X);
D.set_node(1, X);
D.set_edge(0, 0, 1, F); // edge 0: (0,1)

```

We can then apply the Chain and Hom functors, to obtain a diagram of homology spaces and maps between them

```

using F3 = ModP<int, 3>;
auto CD = bats::Chain(D, F3());

auto HD = bats::Hom(CD, (size_t) 1); // homology in dimension 1

```

Finally, we extract the barcode from the homology diagram

```

auto ps = barcode(HD, 1);
for (auto p : ps) {
    std::cout << p.str() << std::endl;
}

```

The output should look like: $1 : (0,1) <0,0>$. This indicates there is a 1-dimensional homology bar, which is born in the space with index 0 and survives until the space with index 1. The $<0,0>$ indicates which generators are associated with the homology class in the diagram.

1.2.5 Source Code

```

1  #include <vector>
2  #include <iostream>
3  #include <bats.hpp>
4
5  int main() {
6
7      // Simplicial complexes and Homology
8      {
9          bats::SimplicialComplex X;
10         X.add_recursive({0,1,2});
11         X.add_recursive({2,3});
12         X.add({1,3});
13
14         X.print_summary();
15
16         using F2 = ModP<int, 2>;
17         auto R = bats::Reduce(X, F2());
18
19         R.print_summary();
20     }
21
22     // Persistent homology
23     {
24         bats::Filtration<double, bats::SimplicialComplex> F;
25         std::vector<size_t> spx;
26         spx = {0,1,2}; F.add_recursive(0.0, spx);
27         spx = {2,3};   F.add_recursive(1.0, spx);
28         spx = {1,3};   F.add(2.0, spx);
29
30         F.complex().print_summary();
31
32         using F2 = ModP<int, 2>;
33         auto R = bats::Reduce(F, F2());
34
35         for (size_t k = 0; k < R.maxdim(); ++k) {
36             std::cout << "\n" << k << "-dimensional barcode:\n";
37             for (auto p : R.persistence_pairs(k)) {
38                 std::cout << p.str() << std::endl;
39             }
40         }
41     }
42
43     // Maps
44     {
45         bats::SimplicialComplex X;
46         X.add_recursive({0,1});
47         X.add_recursive({1,2});
48         X.add_recursive({2,3});
49         X.add_recursive({0,3});
50         bats::SimplicialComplex Y = X; // copy
51

```

(continues on next page)

(continued from previous page)

```

52     X.print_summary();
53
54     std::vector<size_t> f0{2,1,0,3}; // reflection map
55     auto F = bats::SimplicialMap(X, Y, f0);
56
57     // apply the chain functor
58     using F3 = ModP<int, 3>;
59     auto CX = bats::Chain(X, F3());
60     auto CY = bats::Chain(Y, F3());
61     auto CF = bats::Chain(F, F3());
62
63     auto RX = bats::ReducedChainComplex(CX);
64     auto RY = bats::ReducedChainComplex(CY);
65     RX.print_summary();
66     RY.print_summary();
67
68     for (size_t k = 0; k < 2; k++) {
69         std::cout << "\nInduced map in dimension " << k << std::endl;
70         auto HF = bats::induced_map(CF, RX, RY, k);
71         HF.print();
72     }
73 }
74
75 // Zigzag homology
76 std::cout << "\nZigzag Example\n";
77 {
78     bats::Diagram<bats::SimplicialComplex, bats::CellularMap> D(2,1);
79
80     bats::SimplicialComplex X;
81     X.add_recursive({0,1});
82     X.add_recursive({1,2});
83     X.add_recursive({2,3});
84     X.add_recursive({0,3});
85
86     std::vector<size_t> f0{2,1,0,3}; // reflection map
87     auto F = bats::SimplicialMap(X, X, f0);
88
89     D.set_node(0, X);
90     D.set_node(1, X);
91     D.set_edge(0, 0, 1, F); // edge 0: (0,1)
92
93     using F3 = ModP<int, 3>;
94     auto CD = bats::Chain(D, F3());
95
96     auto HD = bats::Hom(CD, (size_t) 1); // homology in dimension 1
97
98     auto ps = bats::barcode(HD, 1);
99     for (auto p : ps) {
100         std::cout << p.str() << std::endl;
101     }
102 }
103

```

(continues on next page)

(continued from previous page)

```
104     }  
105  
106  
107     return EXIT_SUCCESS;  
108 }
```

1.3 Tutorials

1.3.1 Complexes

This tutorial will cover building and using complexes which represent topological spaces.

Interfaces

Chain Functor

There are several functions a complex should implement in order to be compatible with the chain functor.

Filtrations

Filtrations can be created from a complex as well as a list of filtration values. However, in order to construct filtrations incrementally the following methods should be implemented.

1.4 Examples

1.4.1 Rips Complexes

A Rips complex is constructed on a finite metric space with a fixed distance parameter. All pairs of points which have a smaller distance than the specified parameter are connected by an edge, and the maximal simplicial complex (up to a specified dimension) is constructed on this edge set (i.e. a flag complex or clique complex).

First, let's generate a dataset:

Standard Rips Complexes

now we can construct a Rips complex. It can either be a `SimplicialComplex` or `LightSimplicialComplex`.

```
// using CpxT = SimplicialComplex;  
using CpxT = LightSimplicialComplex<size_t, std::unordered_map<size_t, size_t>>;  
  
size_t maxdim = 3; // maximum simplex dimension  
double rmax = 0.2; // connectivity radius  
  
auto X = RipsComplex<CpxT>(x, LInfDist(), rmax, maxdim);
```

Rips Filtrations

A Rips filtration can be constructed with an almost identical call. The difference is that now you keep track of the parameter at which each simplex appears in the filtration and can compute persistent homology.

```
auto F = bats::RipsFiltration<CpxT>(X, dist, rmax, maxdim);
```

1.4.2 Zigzag Filtrations

A zigzag filtration allows cells to enter and exit a complex at specified parameters. Construction is similar to a filtration, but you need to specify entry as well as exit time. Zigzag filtration functionality is under the `bats::zigzag` namespace.

```
bats::zigzag::ZigzagFiltration<bats::SimplicialComplex> F;

std::vector<size_t> spx;
// create a cycle that persists for a while
spx = {0,1}; F.add_recursive(0.0, 10.0, spx);
spx = {0,2}; F.add_recursive(0.0, 10.0, spx);
spx = {1,2}; F.add_recursive(0.0, 10.0, spx);

// now block cycle for some period of time
spx = {0,1,2}; F.add(2.0, 4.0, spx);
```

You can compute a barcode using `bats::barcode`

```
using F2 = ModP<int, 2>;
auto ps = bats::zigzag::barcode(F, 1, F2(),
    bats::no_optimization_flag(),
    bats::standard_reduction_flag()
);

for (auto& pk : ps) {
    for (auto p : pk) {
        if (p.length() > 0)
            std::cout << p << std::endl;
    }
}
```

You should see the following output

```
0 : (0,10) <0(1),0(0)>
1 : (0,2) <2(1),0(1)>
1 : (4,10) <0(0),2(0)>
```

This indicates that there is a single connected component in the zigzag filtration. The two H_1 classes correspond to the cycle, which isn't present when we put in the triangle between parameters 2 and 4.

1.5 Library API

1.5.1 Full API

Namespaces

Namespace bats

Contents

- *Detailed Description*
- *Namespaces*
- *Classes*
- *Functions*
- *Typedefs*
- *Variables*

Detailed Description

We need to specify what is the conventional usage of permutation in bats!! [2, 0, 1] apply to [2.0 , 1.0 ,5.0] a) is [5.0, 2.0, 1.0] in matrix permutation notation b) is [1.0, 5.0 ,2.0] in traditioanl notation(in book From Mathematics to Generic Programming) We take the frist notation here, but notice that BATs, sometimes, will mix the two notations!!! Check before use!!! Since the above two notations are inverse to each other, inverse them if needed! Constructions on a grid Freudenthal triangulation

Namespaces

- *Namespace bats::detail*
- *Namespace bats::flags*
- *Namespace bats::future*
- *Namespace bats::rowmajor*
- *Namespace bats::util*
- *Namespace bats::zigzag*

Classes

- *Template Struct AbstractMetric*
- *Struct AngleDist*
- *Struct apparent_pairs_flag*
- *Struct bar*
- *Template Struct ChainComplex*
- *Template Struct ChainMap*
- *Struct clearing_flag*
- *Template Struct CochainComplex*
- *Struct compression_flag*
- *Struct compute_basis_flag*
- *Struct CosineDist*
- *Template Struct DataSet*
- *Template Struct DGLinearMap*
- *Template Struct DGVectorSpace*
- *Struct Diagram::Edge*
- *Struct divide_conquer_flag*
- *Struct Euclidean*
- *Struct extra_reduction_flag*
- *Template Struct filtered_edge*
- *Template Struct FilteredChainComplex*
- *Template Struct FilteredCochainComplex*
- *Template Struct FilteredDGVectorSpace*
- *Struct L1Dist*
- *Struct LightSimplicialComplex::simplex_boundary_iterator*
- *Struct LInfDist*
- *Struct no_apparent_pairs_flag*
- *Struct no_optimization_flag*
- *Template Struct PersistencePair*
- *Template Struct ReducedDGVectorSpace*
- *Template Struct ReducedFilteredChainComplex*
- *Template Struct ReducedFilteredDGVectorSpace*
- *Struct RPAngleDist*
- *Struct RPCosineDist*
- *Struct sparse_reduction_flag*
- *Struct standard_reduction_flag*

- *Template Struct tedge*
- *Struct triangle*
- *Template Struct Update_info*
- *Struct UpdateInfo2*
- *Template Class BarcodePair*
- *Class CellComplex*
- *Class CellularMap*
- *Class CubicalComplex*
- *Template Class Diagram*
- *Template Class Filtration*
- *Template Class LightSimplicialComplex*
- *Template Class ReducedChainComplex*
- *Template Class ReducedCochainComplex*
- *Class SimplicialComplex*
- *Template Class SparseTrie*

Functions

- *Template Function bats::__ChainComplex(const CpxT&, T)*
- *Template Function bats::__ChainComplex(const CpxT&, const CpxT&, T)*
- *Template Function bats::__ChainFunctor*
- *Template Function bats::__CochainComplex*
- *Template Function bats::__FilteredChainComplex*
- *Template Function bats::__ReducedChainComplex*
- *Template Function bats::__ReducedCochainComplex*
- *Template Function bats::__ReducedFilteredChainComplex*
- *Template Function bats::add_dimension_recursive_flag(Filtration<T, CpxT>&, const NT&, const size_t, const size_t, const std::vector<size_t>&, std::vector<size_t>&, const T&)*
- *Template Function bats::add_dimension_recursive_flag(Filtration<T, CpxT>&, const NT&, const size_t, const size_t, const std::vector<size_t>&, std::vector<size_t>&, const T&, bool, const cell_ind&)*
- *Template Function bats::add_dimension_recursive_flag(CpxT&, const NT&, const size_t, const size_t, const std::vector<size_t>&, std::vector<size_t>&)*
- *Template Function bats::add_dimension_recursive_flag_extension*
- *Template Function bats::add_dimension_recursive_flag_unsafe*
- *Function bats::add_dimension_recursive_nerve*
- *Template Function bats::add_normal_noise(Matrix<T>&, unsigned, const T, const T)*
- *Template Function bats::add_normal_noise(DataSet<T>&, const T, const T)*
- *Template Function bats::add_normal_noise(DataSet<T>&, unsigned, const T, const T)*

- *Template Function* `bats::add_normal_noise(Matrix<T>&, const T, const T)`
- *Template Function* `bats::add_uniform_noise(Matrix<T>&, unsigned, const T, const T)`
- *Template Function* `bats::add_uniform_noise(Matrix<T>&, const T, const T)`
- *Template Function* `bats::add_uniform_noise(DataSet<T>&, const T, const T)`
- *Function* `bats::all_pairs`
- *Template Function* `bats::approx_center`
- *Template Function* `bats::assign_set_lower`
- *Template Function* `bats::assign_set_upper`
- *Template Function* `bats::barcode(const Diagram<NT, TM>&, size_t)`
- *Template Function* `bats::barcode(const Diagram<NT, TM>&, size_t, flags::divide_conquer)`
- *Template Function* `bats::barcode(const Diagram<NT, TM>&, size_t, flags::leftward)`
- *Template Function* `bats::barcode(const Diagram<NT, TM>&, size_t, flags::rightward)`
- *Template Function* `bats::barcode(const Diagram<NT, std::vector<TM>>&, Args...)`
- *Template Function* `bats::barcode_equality`
- *Template Function* `bats::barcode_form_divide_conquer`
- *Template Function* `bats::barcode_form_leftright`
- *Template Function* `bats::barcode_form_rightleft`
- *Template Function* `bats::barcode_from_barcode_form`
- *Function* `bats::bars_to_pairs`
- *Template Function* `bats::binom_to_inds`
- *Function* `bats::bivariate_cover`
- *Template Function* `bats::Chain(const CpxT&, T)`
- *Template Function* `bats::Chain(const CpxT&, const CpxT&, T)`
- *Template Function* `bats::Chain(const CellularMap&, T)`
- *Template Function* `bats::Chain(const Filtration<FT, CpxT>&, T)`
- *Template Function* `bats::Chain(const Diagram<CpxT, CellularMap>&, T)`
- *Template Function* `bats::ChainFunctor(const DT&)`
- *Template Function* `bats::ChainFunctor(const DT&, TF)`
- *Template Function* `bats::circle`
- *Template Function* `bats::CompleteFlagFiltration`
- *Template Function* `bats::coordinate_projection`
- *Function* `bats::Cube(size_t, size_t, size_t, size_t, size_t, size_t, size_t, size_t, size_t)`
- *Function* `bats::Cube(size_t, size_t)`
- *Function* `bats::Cube(size_t, size_t, size_t)`
- *Function* `bats::CubicalMap`
- *Template Function* `bats::DGLinearFunctor`

- *Template Function* `bats::DiscreteMorozovZigzag`
- *Template Function* `bats::DiscreteMorozovZigzagSets`
- *Template Function* `bats::dowker_edge_param(const Matrix<T>&, const size_t, const size_t)`
- *Template Function* `bats::dowker_edge_param(const Matrix<T>&)`
- *Template Function* `bats::dowker_filtration_edges`
- *Template Function* `bats::DowkerFiltration(const Matrix<T>&, T, size_t)`
- *Template Function* `bats::DowkerFiltration(const DataSet<T>&, const DataSet<T>&, const M&, T, size_t)`
- *Template Function* `bats::DowkerFiltration(const Matrix<T>&, const bats::Cover&, T, size_t)`
- *Template Function* `bats::DowkerFiltration(const DataSet<T>&, const DataSet<T>&, const M&, const bats::Cover&, T, size_t)`
- *Template Function* `bats::EilenbergZilber(const CpxT&, const CpxT&, const size_t, T)`
- *Template Function* `bats::EilenbergZilber(const CpxT&, const CpxT&, const CpxT&, const CpxT&, const size_t, T)`
- *Template Function* `bats::enclosing_radius`
- *Template Function* `bats::euclidean`
- *Function* `bats::extension_perm`
- *Template Function* `bats::extract_basis_indices(const MT&, const std::vector<size_t>&)`
- *Template Function* `bats::extract_basis_indices(const MT&)`
- *Template Function* `bats::extract_dimension`
- *Template Function* `bats::extract_pairs`
- *Function* `bats::filtration_iperm`
- *Template Function* `bats::filtration_sortperm(const std::vector<T>&)`
- *Template Function* `bats::filtration_sortperm(const std::vector<std::vector<T>>&)`
- *Function* `bats::find_parent`
- *Template Function* `bats::find_perm_from_vector`
- *Template Function* `bats::flag_filtration_edges`
- *Template Function* `bats::FlagComplex`
- *Template Function* `bats::FlagFiltration`
- *Template Function* `bats::FlagFiltration_extension`
- *Template Function* `bats::force_repel_rp`
- *Template Function* `bats::Freudenthal(const CubicalComplex&, size_t, size_t, size_t)`
- *Template Function* `bats::Freudenthal(size_t, size_t, size_t, size_t, size_t, size_t, size_t, size_t, size_t, size_t)`
- *Template Function* `bats::Freudenthal(size_t, size_t)`
- *Template Function* `bats::Freudenthal(size_t, size_t, size_t)`
- *Function* `bats::gen_cylinder`
- *Function* `bats::get_clearing_inds`
- *Template Function* `bats::get_compression_inds`

- *Template Function* `bats::get_dM_ZZ_inds`
- *Template Function* `bats::get_m`
- *Template Function* `bats::get_subset`
- *Template Function* `bats::greedy_landmarks`
- *Template Function* `bats::greedy_landmarks_hausdorff(const DataSet<T>&, const M&, const size_t, const size_t)`
- *Template Function* `bats::greedy_landmarks_hausdorff(const Matrix<T>&, const size_t)`
- *Template Function* `bats::greedy_landmarks_hausdorff(const DataSet<T>&, const M&, const size_t)`
- *Template Function* `bats::hausdorff_landmarks`
- *Template Function* `bats::Hom(const Diagram<ChainComplex<TM>, ChainMap<TM>>&, size_t)`
- *Template Function* `bats::Hom(const Diagram<ChainComplex<TM>, ChainMap<TM>>&, bool)`
- *Template Function* `bats::Hom(const Diagram<DGVectorSpace<TM>, DGLinearMap<TM>>&, size_t)`
- *Template Function* `bats::Hom(const Diagram<DGVectorSpace<TM>, DGLinearMap<TM>>&, bool)`
- *Function* `bats::identity_perm`
- *Template Function* `bats::increment_m`
- *Template Function* `bats::induced_map(const ChainMap<ColumnMatrix<TVec>>&, const ReducedChainComplex<ColumnMatrix<TVec>>&, const ReducedChainComplex<ColumnMatrix<TVec>>&, size_t)`
- *Template Function* `bats::induced_map(const ChainMap<ColumnMatrix<TVec>>&, const ReducedCochainComplex<ColumnMatrix<TVec>>&, const ReducedCochainComplex<ColumnMatrix<TVec>>&, size_t)`
- *Template Function* `bats::induced_map(const DGLinearMap<ColumnMatrix<TVec>>&, const ReducedDGVectorSpace<ColumnMatrix<TVec>>&, const ReducedDGVectorSpace<ColumnMatrix<TVec>>&, size_t)`
- *Template Function* `bats::interval`
- *Template Function* `bats::is_left_arrow`
- *Template Function* `bats::k_choose_2`
- *Function* `bats::k_choose_2_inv`
- *Template Function* `bats::kdist`
- *Template Function* `bats::Kendall_tau(const std::vector<T>&)`
- *Template Function* `bats::Kendall_tau(const std::vector<T>&, const std::vector<T>&)`
- *Template Function* `bats::Kendall_tau(const std::vector<std::vector<T>>&)`
- *Template Function* `bats::Kendall_tau_inplace`
- *Template Function* `bats::Kendall_tau_merge`
- *Template Function* `bats::kron_chain_shift`
- *Template Function* `bats::kron_chains`
- *Template Function* `bats::kron_homology`
- *Template Function* `bats::kron_index`
- *Template Function* `bats::landmark_cover`
- *Template Function* `bats::landmark_eps_cover`

- Function `bats::linear_cover_intersection_diagram`
- Function `bats::linear_subset_union_diagram`
- Template Function `bats::LowerStarFiltration`
- Template Function `bats::make_edge`
- Template Function `bats::mayer_vietoris_boundary`
- Template Function `bats::neighborhood(const VectorView<T>&, const DataSet<T>&, const M&, const T)`
- Template Function `bats::neighborhood(const VectorView<T>&, const DataSet<T>&, const M&, const size_t)`
- Template Function `bats::neighborhoods(const DataSet<T>&, const DataSet<T>&, const M&, const T)`
- Template Function `bats::neighborhoods(const DataSet<T>&, const DataSet<T>&, const M&, const size_t)`
- Template Function `bats::neighborhoods(const Matrix<T>&, const T)`
- Function `bats::Nerve(const bats::Cover&, const size_t)`
- Function `bats::Nerve(const Diagram<bats::Cover, std::vector<size_t>>&, const size_t)`
- Template Function `bats::normalize_entries`
- Template Function `bats::operator<`
- Template Function `bats::operator<<`
- Template Function `bats::OscillatingRipsZigzagSets`
- Template Function `bats::pairwise_dist`
- Template Function `bats::partial_reduce_parallel(ColumnMatrix<TVec>&, const size_t)`
- Template Function `bats::partial_reduce_parallel(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&, const size_t)`
- Template Function `bats::pass_L_left`
- Template Function `bats::pass_P_left`
- Template Function `bats::pass_P_right`
- Template Function `bats::pass_PL_left`
- Template Function `bats::pass_U_right`
- Template Function `bats::pass_UP_right`
- Function `bats::perm_to_the_end(const size_t&, const size_t&)`
- Function `bats::perm_to_the_end(const std::vector<size_t>&, const size_t&)`
- Template Function `bats::print_1D_vectors`
- Template Function `bats::print_2D_vectors`
- Template Function `bats::print_filtration_info`
- Template Function `bats::print_simplex`
- Template Function `bats::print_summary_of_filtration`
- Function `bats::prod_ind`
- Template Function `bats::product_paths(CpxT&, itT, const itT, itT, const itT, std::vector<size_t>&, const size_t)`
- Template Function `bats::product_paths(CpxT&, itT, const itT, itT, const itT, std::vector<size_t>&, const size_t, std::vector<cell_ind>&)`

- Template Function `bats::product_space`
- Template Function `bats::random_landmarks`
- Template Function `bats::read_point_cloud`
- Template Function `bats::Reduce(const Filtration<FT, CpxT>&, T, Args ...)`
- Template Function `bats::Reduce(const FilteredChainComplex<T, MT>&, Args ...)`
- Template Function `bats::Reduce(const CpxT&, T, Args ...)`
- Template Function `bats::Reduce(const ChainComplex<MT>&, Args ...)`
- Template Function `bats::reduce_block_dq(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&, const size_t, const size_t, const size_t)`
- Template Function `bats::reduce_block_dq(ColumnMatrix<TVec>&, const size_t, const size_t, const size_t, const size_t, const size_t)`
- Template Function `bats::reduce_block_sequential(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&, const size_t, const size_t)`
- Template Function `bats::reduce_block_sequential(ColumnMatrix<TVec>&, const size_t, const size_t)`
- Template Function `bats::reduce_column_extra`
- Template Function `bats::reduce_column_sparsify`
- Template Function `bats::reduce_column_standard(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&, const size_t, std::unordered_map<size_t, size_t>&, typename TVec::tmp_type&)`
- Template Function `bats::reduce_column_standard(ColumnMatrix<TVec>&, const size_t, std::unordered_map<size_t, size_t>&, typename TVec::tmp_type&)`
- Template Function `bats::reduce_matrix(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&)`
- Template Function `bats::reduce_matrix(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&, bats::standard_reduction_flag)`
- Template Function `bats::reduce_matrix(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&, bats::extra_reduction_flag)`
- Template Function `bats::reduce_matrix(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&, divide_conquer_flag)`
- Template Function `bats::reduce_matrix(ColumnMatrix<TVec>&, divide_conquer_flag)`
- Template Function `bats::reduce_matrix(ColumnMatrix<TVec>&)`
- Template Function `bats::reduce_matrix(ColumnMatrix<TVec>&, bats::standard_reduction_flag)`
- Template Function `bats::reduce_matrix(ColumnMatrix<TVec>&, bats::extra_reduction_flag)`
- Template Function `bats::reduce_matrix_clearing(ColumnMatrix<TVec>&, const std::vector<size_t>&, flag)`
- Template Function `bats::reduce_matrix_clearing(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&, const ColumnMatrix<TVec>&, const std::vector<size_t>&, flag)`
- Template Function `bats::reduce_matrix_compression(ColumnMatrix<TVec>&, const std::vector<bool>&, flag)`
- Template Function `bats::reduce_matrix_compression(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&, const std::vector<bool>&, flag)`
- Template Function `bats::reduce_matrix_extra(ColumnMatrix<TVec>&)`
- Template Function `bats::reduce_matrix_extra(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&)`

- Template Function `bats::reduce_matrix_standard(ColumnMatrix<TVec>&)`
- Template Function `bats::reduce_matrix_standard(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&)`
- Template Function `bats::remove_extra_cycles`
- Template Function `bats::Rips(const Diagram<std::set<size_t>, std::vector<size_t>>&, const DataSet<T>&, const M&, const T, const size_t)`
- Template Function `bats::Rips(const Diagram<std::set<size_t>, std::vector<size_t>>&, const DataSet<T>&, const M&, const std::vector<T>&, const size_t)`
- Template Function `bats::rips_edges(const DataSet<T>&, const M&, const T)`
- Template Function `bats::rips_edges(const Matrix<T>&, const T)`
- Template Function `bats::rips_edges(std::vector<T>&, std::vector<size_t>&, std::vector<T>&)`
- Template Function `bats::rips_filtration_edges(const DataSet<T>&, const M&, const T)`
- Template Function `bats::rips_filtration_edges(const Matrix<T>&, const T)`
- Template Function `bats::rips_filtration_edges(const DataSet<T>&, const bats::Cover&, const M&, const T)`
- Template Function `bats::rips_union_find_pairs`
- Template Function `bats::RipsComplex(const DataSet<T>&, const M&, T, size_t)`
- Template Function `bats::RipsComplex(const Matrix<T>&, T, size_t)`
- Template Function `bats::RipsFiltration(const DataSet<T>&, const M&, T, size_t)`
- Template Function `bats::RipsFiltration(const Matrix<T>&, T, size_t)`
- Template Function `bats::RipsFiltration(const DataSet<T>&, const bats::Cover&, const M&, T, size_t)`
- Template Function `bats::RipsFiltration_extension(const DataSet<T>&, const M&, T, size_t)`
- Template Function `bats::RipsFiltration_extension(const Matrix<T>&, T, size_t)`
- Template Function `bats::RipsHom`
- Template Function `bats::sample_cube(const size_t, const size_t)`
- Template Function `bats::sample_cube(const size_t, const size_t, unsigned)`
- Template Function `bats::sample_sphere(const size_t, const size_t, unsigned)`
- Template Function `bats::sample_sphere(const size_t, const size_t)`
- Function `bats::serpinski_diagram`
- Template Function `bats::SimplicialMap(const CpxT&, const CpxT&)`
- Template Function `bats::SimplicialMap(const CpxT&, const CpxT&, const std::vector<size_t>&)`
- Template Function `bats::sort_edges`
- Template Function `bats::sort_indexes`
- Template Function `bats::sparsify_basis(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&, const size_t, std::map<F, size_t>&, typename TVec::tmp_type&)`
- Template Function `bats::sparsify_basis(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&)`
- Template Function `bats::StrictRipsCoverFiltration`
- Template Function `bats::StrictWitnessComplex`
- Template Function `bats::test_reduce_result`

- *Template Function* `bats::TriangulatedProduct(const CpxT&, const CpxT&)`
- *Template Function* `bats::TriangulatedProduct(const CpxT&, const CpxT&, const size_t, const size_t)`
- *Template Function* `bats::TriangulatedProduct(const CpxT&, const CpxT&, const size_t)`
- *Template Function* `bats::type_A_dq_common`
- *Template Function* `bats::type_A_dq_EL`
- *Template Function* `bats::type_A_dq_EU`
- *Template Function* `bats::type_A_leftright_sweep1`
- *Template Function* `bats::type_A_leftright_sweep2`
- *Template Function* `bats::type_A_rightleft_sweep1`
- *Template Function* `bats::type_A_rightleft_sweep2`
- *Template Function* `bats::uniform_interval_cover`
- *Template Function* `bats::union_find_pairs`
- *Template Function* `bats::vertex_inclusion_map`
- *Template Function* `bats::witness_edge_param`
- *Template Function* `bats::witness_edges(const DataSet<T>&, const DataSet<T>&, const M&)`
- *Template Function* `bats::witness_edges(const DataSet<T>&, const DataSet<T>&, const M&, const T)`
- *Template Function* `bats::witness_neighborhoods`
- *Template Function* `bats::WitnessComplex(const DataSet<T>&, const DataSet<T>&, const M&, const size_t)`
- *Template Function* `bats::WitnessComplex(const DataSet<T>&, const DataSet<T>&, const M&, const size_t, const T, const size_t)`
- *Template Function* `bats::WitnessFiltration`
- *Template Function* `bats::zigzag_levelsets`
- *Template Function* `bats::zigzag_toplex`

Typedefs

- *Typedef* `bats::Cover`
- *Typedef* `bats::Matrix`

Variables

- *Variable* `bats::NO_IND`

Namespace `bats::detail`

Contents

- *Functions*

Functions

- *Template Function `bats::detail::pivot_coeff`*

Namespace `bats::flags`

Contents

- *Classes*

Classes

- *Struct `divide_conquer`*
- *Struct `leftward`*
- *Struct `rightward`*

Namespace `bats::future`

Contents

- *Classes*
- *Functions*

Classes

- *Struct `ColumnMajor`*
- *Struct `ElementaryPermutation`*
- *Template Struct `LUFact`*
- *Struct `RowMajor`*
- *Template Struct `SimilarityTransform`*
- *Class `CompositePermutation`*
- *Template Class `const_strided_iterator`*

- *Template Class Matrix*
- *Template Class MatrixView*
- *Class Permutation*
- *Template Class range*
- *Class range::const_iterator*
- *Template Class Span*
- *Template Class strided_iterator*
- *Template Class VectorView*

Functions

- *Template Function bats::future::find_pivot_complete*
- *Template Function bats::future::find_pivot_high(const RandomAccessIterator&, ssize_t, ssize_t)*
- *Template Function bats::future::find_pivot_high(const RandomAccessIterator&, ssize_t)*
- *Template Function bats::future::gemm*
- *Template Function bats::future::gemv(const MT&, const VT&)*
- *Template Function bats::future::gemv(const MT&, const VT1&, VT2&&)*
- *Template Function bats::future::l_residual(const MT&, I1&)*
- *Template Function bats::future::l_residual(const MT&&, I1&&)*
- *Template Function bats::future::l_solve(const MT&, I1&, I2&)*
- *Template Function bats::future::l_solve(const MT1&, const MT2&, I1&, I2&)*
- *Template Function bats::future::LU*
- *Function bats::future::operator<<(std::ostream&, const CompositePermutation&)*
- *Function bats::future::operator<<(std::ostream&, const ElementaryPermutation&)*
- *Template Function bats::future::unit_lower*
- *Template Function bats::future::upper*

Namespace bats::rowmajor

Contents

- *Functions*

Functions

- *Template Function* `bats::rowmajor::get_idx(T, T, T)`
- *Template Function* `bats::rowmajor::get_idx(T, T, T, T, T)`

Namespace `bats::util`

Contents

- *Namespaces*
- *Classes*
- *Functions*

Namespaces

- *Namespace* `bats::util::io`

Classes

- *Struct* `SimplexHasher`
- *Class* `SimplexContainer`

Functions

- *Template Function* `bats::util::apply_iperm_swap`
- *Template Function* `bats::util::apply_perm(T, std::vector<T2>&, const std::vector<size_t>&)`
- *Template Function* `bats::util::apply_perm(T *, const std::vector<size_t>&)`
- *Template Function* `bats::util::apply_perm(std::vector<T>&, const std::vector<size_t>&)`
- *Template Function* `bats::util::apply_perm_swap`
- *Function* `bats::util::fill_partial_sortperm`
- *Template Function* `bats::util::fill_sortperm(const T&, const T&, std::vector<size_t>&)`
- *Template Function* `bats::util::fill_sortperm(const std::vector<T>&, std::vector<size_t>&)`
- *Template Function* `bats::util::find_sorted_lt`
- *Template Function* `bats::util::firstk`
- *Template Function* `bats::util::has_intersect_sorted`
- *Template Function* `bats::util::has_intersect_sorted_lt`
- *Template Function* `bats::util::intersect_sorted(const C1&, const C2&, std::vector<T>&)`
- *Template Function* `bats::util::intersect_sorted(const C1&, const C2&, std::set<T>&)`
- *Template Function* `bats::util::intersect_sorted_lt`

- Function `bats::util::inv_perm`
- Template Function `bats::util::is_degenerate`
- Function `bats::util::partial_perm`
- Template Function `bats::util::perm_inversions`
- Template Function `bats::util::perm_sign`
- Function `bats::util::rand_perm`
- Function `bats::util::random_subset`
- Template Function `bats::util::read_simplex`
- Template Function `bats::util::set_intersection`
- Template Function `bats::util::set_union`
- Template Function `bats::util::simplex_sign`
- Function `bats::util::sort_ind_by_perm`
- Function `bats::util::sort_ind_pair_by_perm`
- Template Function `bats::util::sort_into`
- Template Function `bats::util::sort_sum_reduce`
- Template Function `bats::util::sorted_complement`
- Template Function `bats::util::sortperm(const std::vector<T>&)`
- Template Function `bats::util::sortperm(const TI&, const TI&)`
- Template Function `bats::util::sortperm(RAI, RAI, Compare)`
- Template Function `bats::util::sortperm(const size_t, const size_t, Compare)`
- Template Function `bats::util::stable_sortperm`
- Template Function `bats::util::to_set`
- Template Function `bats::util::top_k`
- Template Function `bats::util::top_p`
- Template Function `bats::util::write_simplex(IO&, std::vector<T>&)`
- Template Function `bats::util::write_simplex(IO&, TI&&, TI&&)`

Namespace `bats::util::io`

Contents

- *Functions*

Functions

- *Template Function* `bats::util::io::parse_argv(const int, char **, const std::string&&, const T)`
- *Function* `bats::util::io::parse_argv(const int, char **, const std::string&&, const std::string)`

Namespace `bats::zigzag`

Contents

- *Namespaces*
- *Classes*
- *Functions*

Namespaces

- *Namespace* `bats::zigzag::detail`

Classes

- *Template Struct* `rfilt_val`
- *Template Struct* `ZigzagChainComplex`
- *Template Struct* `ZigzagPair`
- *Template Class* `ZigzagFiltration`

Functions

- *Template Function* `bats::zigzag::barcode`
- *Template Function* `bats::zigzag::extend_levelset`
- *Template Function* `bats::zigzag::extend_zigzag_filtration(const std::vector<T>&, const CubicalComplex&, const T, const size_t)`
- *Template Function* `bats::zigzag::extend_zigzag_filtration(const std::vector<T>&, const CpxT&, const T)`
- *Template Function* `bats::zigzag::extra_col_reduction`
- *Template Function* `bats::zigzag::prepare_ChainComplex(const ZigzagFiltration<CpxT, T>&, FT)`
- *Template Function* `bats::zigzag::prepare_ChainComplex(const ZigzagFiltration<bats::SimplicialComplex, T>&, FT)`
- *Template Function* `bats::zigzag::reduce_column(size_t, ColumnMatrix<VecT>&, ColumnMatrix<VecT>&, std::vector<size_t>&, typename VecT::tmp_type&, bats::extra_reduction_flag)`
- *Template Function* `bats::zigzag::reduce_column(size_t, ColumnMatrix<VecT>&, ColumnMatrix<VecT>&, std::vector<size_t>&, typename VecT::tmp_type&, reduction_flag)`
- *Template Function* `bats::zigzag::zigzag_barcode_reduction`

Namespace `bats::zigzag::detail`

Contents

- *Functions*

Functions

- *Template Function `bats::zigzag::detail::apply_basis`*
- *Template Function `bats::zigzag::detail::boundary_insertion_map`*
- *Template Function `bats::zigzag::detail::cube_extrema`*
- *Template Function `bats::zigzag::detail::cube_val`*
- *Template Function `bats::zigzag::detail::cycle_insertion_map`*
- *Function `bats::zigzag::detail::lex_cmp`*
- *Function `bats::zigzag::detail::rlex_cmp`*
- *Template Function `bats::zigzag::detail::simplex_extrema`*
- *Template Function `bats::zigzag::detail::update_bars`*

Classes and Structs

Template Struct A

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_matrix_interface.hpp`

Inheritance Relationships

Base Type

- `public MAT` (*Struct MAT*)

Derived Types

- `public E< Impl >` (*Template Struct E*)
- `public T< Impl >` (*Template Struct T*)

Struct Documentation

```
template<typename Impl>
struct A : public MAT
    Subclassed by E< Impl >, T< Impl >
```

Template Struct AbstractMetric

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_metric.hpp`

Struct Documentation

```
template<class D>
struct AbstractMetric
```

Public Functions

```
template<typename T>
inline T dist(const VectorView<T> &x, const VectorView<T> &y) const
```

```
template<typename T>
inline T operator()(const VectorView<T> &x, const VectorView<T> &y) const
```

```
template<typename T>
inline std::vector<T> operator()(const VectorView<T> &x, const DataSet<T> &X) const
```

```
template<typename T>
inline Matrix<T> operator()(const DataSet<T> &X, const DataSet<T> &Y) const
```

```
template<typename T>
inline Matrix<T> operator()(const DataSet<T> &X) const
```

Struct AngleDist

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_metric.hpp`

Inheritance Relationships

Base Type

- `public bats::AbstractMetric< AngleDist > (Template Struct AbstractMetric)`

Struct Documentation

struct **AngleDist** : public bats::*AbstractMetric*<*AngleDist*>

Public Functions

```
template<typename T>
inline T dist(const VectorView<T> &x, const VectorView<T> &y) const
```

Struct `apparent_pairs_flag`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Struct Documentation

struct **apparent_pairs_flag**

Struct `bar`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Struct Documentation

struct **bar**

Public Functions

```
inline bar()
```

```
inline bar(size_t s, size_t si, size_t t, size_t ti)
```

Public Members

size_t **start**
size_t **start_ind**
size_t **end**
size_t **end_ind**

Template Struct ChainComplex

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_chain_chain_complex.hpp`

Struct Documentation

template<typename **MT**>
struct **ChainComplex**

Public Functions

inline **ChainComplex**()

inline **ChainComplex**(size_t maxd)

inline **ChainComplex**(const std::vector<*MT*> &boundary)

template<typename **CpxT**>
inline **ChainComplex**(const *CpxT* &X)

template<typename **CpxT**>
inline **ChainComplex**(const *CpxT* &X, const *CpxT* &A)

inline size_t **maxdim**() const

inline size_t **dim**(size_t k) const

inline size_t **dim**() const

inline bool **is_valid_complex**() const

inline *ChainComplex* **subcomplex**(std::vector<std::vector<size_t>> &inds) const

inline *ChainComplex* **relative_complex**(std::vector<std::vector<size_t>> &inds) const

```

inline MT &operator[] (size_t k)
    reference to k-dimensional boundary

    Parameters k – dimension

inline const MT &operator[] (size_t k) const

inline void permute_basis(size_t k, const std::vector<size_t> &perm)

inline void permute_basis(const std::vector<std::vector<size_t>> &perm)

inline void ipermute_basis(size_t k, const std::vector<size_t> &perm)

inline void ipermute_basis(const std::vector<std::vector<size_t>> &perm)

template<typename Information_type>
inline void update_basis_general(size_t k, const Information_type &UI)

template<typename Information_type>
inline void update_basis_general(const Information_type &UI_fast)

inline void clear_compress_apparent_pairs()
    A preprocessing step for computing homology using the reduction algorithm.

```

Warning: using this function will invalidate any basis used by a homology algorithm since no basis vector will be obtained for the cleared columns

Public Members

```
std::vector<MT> boundary
```

Friends

```

inline friend ChainComplex tensor_product(const ChainComplex &A, const ChainComplex &B, size_t
    dmax)

inline friend ChainComplex tensor_product(const ChainComplex &A, const ChainComplex &B)

```

Template Struct ChainMap

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_chain_chain_map.hpp`

Struct Documentation

template<typename **TM**>

struct **ChainMap**

Public Functions

inline **ChainMap**()

inline **ChainMap**(const std::vector<*TM*> &map)

inline **ChainMap**(size_t d)

inline **ChainMap**(const *CellularMap* &f)

template<typename **CpxT**>

inline **ChainMap**(const *CellularMap* &f, const *CpxT* &X, const *CpxT* &A, const *CpxT* &Y, const *CpxT* &B)

inline *ChainMap* **relative_map**(std::vector<std::vector<size_t>> &inds1, std::vector<std::vector<size_t>> &inds2) const

inline size_t **maxdim**() const

inline *TM* &**operator**[](size_t k)

inline const *TM* &**operator**[](size_t k) const

inline void **permute_row_basis**(size_t k, const std::vector<size_t> &p)

inline void **permute_column_basis**(size_t k, const std::vector<size_t> &p)

Public Members

`std::vector<TM> map`

Struct `clearing_flag`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Struct Documentation

struct **clearing_flag**

Template Struct `CochainComplex`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_chain_cochain_complex.hpp`

Struct Documentation

template<typename **MT**>

struct **CochainComplex**

Public Functions

inline **CochainComplex**()

inline **CochainComplex**(const std::vector<size_t> &dim, const std::vector<*MT*> &coboundary)

template<typename **CpxT**>

inline **CochainComplex**(const *CpxT* &X)

inline size_t **maxdim**() const

inline *MT* &**operator[]**(size_t k)

inline void **permute_basis**(size_t k, const std::vector<size_t> &perm)

inline void **permute_basis**(const std::vector<std::vector<size_t>> &perm)

inline void **ipermute_basis**(size_t k, const std::vector<size_t> &perm)

Public Members

`std::vector<size_t> dim`

`std::vector<MT> coboundary`

Struct `compression_flag`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Struct Documentation

struct `compression_flag`

Struct `compute_basis_flag`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Struct Documentation

struct `compute_basis_flag`

Struct `CosineDist`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_metric.hpp`

Inheritance Relationships

Base Type

- public `bats::AbstractMetric< CosineDist >` (*Template Struct AbstractMetric*)

Struct Documentation

struct `CosineDist` : public `bats::AbstractMetric<CosineDist>`

Public Functions

```
template<typename T>  
inline T dist(const VectorView<T> &x, const VectorView<T> &y) const
```

Template Struct DataSet

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_data.hpp`

Struct Documentation

```
template<typename T>
```

```
struct DataSet
```

Public Functions

```
inline DataSet()
```

```
inline DataSet(const Matrix<T> &d)
```

```
inline size_t size() const
```

```
inline size_t dim() const
```

```
inline Matrix<T> &get_data()
```

```
template<typename TI>
```

```
inline DataSet operator[] (const TI &inds) const
```

```
inline VectorView<T> operator[] (const int i)
```

```
inline const VectorView<T> operator[] (const int i) const
```

```
inline VectorView<T> operator[] (const size_t i)
```

```
inline const VectorView<T> operator[] (const size_t i) const
```

```
inline T &operator() (const size_t i, const size_t j)
```

```
inline const T &operator() (const size_t i, const size_t j) const
```

Public Members

Matrix<*T*> **data**

Template Struct DGLinearMap

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_dgvs_dgmap.hpp`

Struct Documentation

template<typename **TM**>

struct **DGLinearMap**

A class for a map between two DGVectorSpaces

The map should be between DGVectorSpaces of the same degree and the map should commute with the differential.

If the DGVectorSpaces are augmented, the map should be augmentation preserving.

Public Functions

inline **DGLinearMap**()

inline **DGLinearMap**(const std::vector<*TM*> &map)

inline **DGLinearMap**(size_t d)

inline **DGLinearMap**(const *CellularMap* &f, int deg = -1)

inline ssize_t **maxdim**() const

inline *TM* &**operator**[](ssize_t k)

inline const *TM* &**operator**[](ssize_t k) const

inline void **permute_row_basis**(ssize_t k, const std::vector<size_t> &p)

inline void **permute_column_basis**(ssize_t k, const std::vector<size_t> &p)

Public Members

`std::vector<TM> map`

Template Struct DGVectorSpace

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_dgvs_dgvs.hpp`

Struct Documentation

template<typename MT>

struct **DGVectorSpace**

a class for a differential graded vector space.

This encapsulates both chain and cochain complex constructions

degree is the degree of the differential: -1 for chain complexes (default) +1 for cochain complexes

differential holds the differential maps

We store maps starting on the edge (-1,0) (-1) \rightarrow (0) \rightarrow (1) \rightarrow (2) \rightarrow ... This map is 0 in the case of standard chain/cochain complexes but can be non-zero for augmented chain/cochain complexes

TODO: need to handle +1 boundary in maxdim

Public Functions

inline MT &operator[] (ssize_t k)

Access maps in various dimensions

if degree is +1 (cohomological type), then lowest map index is -1 \rightarrow * \rightarrow * \rightarrow ...

if degree is -1 (homological type) then lowest map index is 0 \leftarrow * \leftarrow * \leftarrow ...

inline const MT &operator[] (ssize_t k) const

inline DGVectorSpace()

inline DGVectorSpace(size_t maxd, int deg = -1)

Construct a DGVector space with maxd dimensions

inline DGVectorSpace(const std::vector<MT> &diff, int deg = -1)

Construct a DGVector space explicitly from differentials

template<typename CpxT>

inline DGVectorSpace(const CpxT &X, const int deg = -1, const bool augmented = false)

inline ssize_t maxdim() const

inline size_t dim(ssize_t k) const

```
inline void permute_basis(ssize_t k, const std::vector<size_t> &perm)
```

```
inline void permute_basis(const std::vector<std::vector<size_t>> &perm)
```

```
inline void ipermute_basis(ssize_t k, const std::vector<size_t> &perm)
```

```
inline void ipermute_basis(const std::vector<std::vector<size_t>> &perm)
```

```
inline bool is_differential(size_t k)  
    checks that  $d_{\{k\text{-degree}\}} d_k = 0$ 
```

Public Members

```
int degree
```

```
std::vector<MT> differential
```

Struct Diagram::Edge

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_multigraph_diagram.hpp`

Nested Relationships

This struct is a nested type of *Template Class Diagram*.

Struct Documentation

```
struct Edge
```

Public Functions

```
inline Edge()
```

```
inline Edge(size_t s, size_t t)
```

Public Members

size_t **src**

size_t **targ**

Struct `divide_conquer_flag`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_parallel.hpp`

Struct Documentation

struct **divide_conquer_flag**

Struct `Euclidean`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_metric.hpp`

Inheritance Relationships

Base Type

- public `bats::AbstractMetric< Euclidean >` (*Template Struct `AbstractMetric`*)

Struct Documentation

struct **Euclidean** : public `bats::AbstractMetric<Euclidean>`

Public Functions

template<typename **T**>

inline **T dist**(const VectorView<**T**> &x, const VectorView<**T**> &y) const

Struct `extra_reduction_flag`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Struct Documentation

struct **extra_reduction_flag**

Template Struct `filtered_edge`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_flag.hpp`

Struct Documentation

template<typename **T**>

struct **filtered_edge**

Public Functions

inline **filtered_edge**()

inline **filtered_edge**(const size_t &s, const size_t &t, const *T* &r)

inline bool **operator**<(const *filtered_edge* &other) const

Public Members

size_t **s**

size_t **t**

T **r**

Template Struct `FilteredChainComplex`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_chain_filtered_chain_complex.hpp`

Struct Documentation

template<typename **FT**, typename **MT**>

struct **FilteredChainComplex**

Public Functions

```
inline FilteredChainComplex()
```

```
template<typename CpxT>
```

```
inline FilteredChainComplex(const Filtration<FT, CpxT> &F)
```

```
inline size_t dim(const size_t k)
```

```
inline const ChainComplex<MT> &complex() const
```

```
inline const std::vector<std::vector<FT>> &vals() const
```

```
inline void update_filtration(const std::vector<std::vector<FT>> newval)
```

```
template<typename I>
```

```
inline void update_filtration_general(const I &updating_information)
```

Public Members

```
std::vector<std::vector<FT>> val
```

```
ChainComplex<MT> C
```

```
std::vector<std::vector<size_t>> perm
```

Template Struct FilteredCochainComplex

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_chain_filtered_cochain_com`

Struct Documentation

```
template<typename FT, typename MT>
```

```
struct FilteredCochainComplex
```

Public Functions

```
inline FilteredCochainComplex()
```

```
template<typename CpxT>
```

```
inline FilteredCochainComplex(const Filtration<FT, CpxT> &F)
```

```
inline size_t dim(const size_t k)
```

```
inline const CochainComplex<MT> &complex() const

inline const std::vector<std::vector<FT>> &vals() const

inline void update_filtration(const std::vector<std::vector<FT>> newval)

template<typename I>
inline void update_filtration_general(const I &updating_information)
```

Public Members

```
std::vector<std::vector<FT>> val
CochainComplex<MT> C
std::vector<std::vector<size_t>> perm
```

Template Struct FilteredDGVectorSpace

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_dgvs_filtered_dgvs.hpp`

Struct Documentation

```
template<typename FT, typename MT>
struct FilteredDGVectorSpace
```

Public Functions

```
inline void _set_perms()
    put underlying complex in permutation order

inline FilteredDGVectorSpace()

template<typename CpxT>
inline FilteredDGVectorSpace(const Filtration<FT, CpxT> &F, int deg = -1, bool augmented = false)

inline FilteredDGVectorSpace(const std::vector<MT> &diff, const std::vector<std::vector<FT>> &val, int
    deg = -1)

inline size_t dim(const size_t k)

inline DGVectorSpace<MT> &complex()

inline const DGVectorSpace<MT> &complex() const
```

```
inline const std::vector<std::vector<FT>> &vals() const
```

```
inline void update_filtration(const std::vector<std::vector<FT>> newval)
```

Public Members

```
std::vector<std::vector<FT>> val
```

```
DGVectorSpace<MT> C
```

```
std::vector<std::vector<size_t>> perm
```

Struct divide_conquer

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Struct Documentation

```
struct divide_conquer
```

Flag to choose a divide and conquer algorithm

Struct leftward

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Struct Documentation

```
struct leftward
```

Flag to choose leftward algorithm

Struct rightward

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Struct Documentation

```
struct rightward
```

Flag to choose rightward algorithm

Struct ColumnMajor

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_dense.hpp`

Struct Documentation

struct **ColumnMajor**

Struct ElementaryPermutation

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_permutation.hpp`

Struct Documentation

struct **ElementaryPermutation**

Public Functions

inline **ElementaryPermutation**()

inline **ElementaryPermutation**(size_t i, size_t j)

template<typename T>
inline *T* &**operator**()(*T* &a) const

template<typename T>
inline *T* &**operator**()(*T* &&a) const

inline *CompositePermutation* &**operator**()(*CompositePermutation* &a) const

Public Members

size_t **i**

size_t **j**

Friends

friend std::ostream &**operator**<<(std::ostream &os, const *ElementaryPermutation* &p)

Template Struct LUFact

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_lu.hpp`

Struct Documentation

template<typename **MT**>

struct **LUFact**

Public Types

using **val_type** = typename *MT*::value_type

Public Functions

inline **LUFact**(*MT* &&P, *MT* &&L, *MT* &&U, *MT* &&Q)

inline *MT* **prod**() const

inline void **print_info**() const

inline void **print**() const

Public Members

MT **P**

MT **L**

MT **U**

MT **Q**

Struct RowMajor

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_dense.hpp`

Struct Documentation

struct **RowMajor**

Template Struct SimilarityTransform

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_similarity.hpp`

Struct Documentation

template<typename T>

struct **SimilarityTransform**

Public Functions

inline **SimilarityTransform**(const *Matrix*<T> &A0)

inline size_t **size**() const

inline *T* **operator**()(size_t i, size_t j) const

inline void **print_info**() const

inline void **print**() const

inline *Matrix*<T> **prod**() const

inline void **swap_rows**(size_t i0, size_t i1)

inline void **swap_columns**(size_t j0, size_t j1)

inline void **add_row**(*T* a, size_t i1, size_t i0)

inline void **scale_row**(*T* a, size_t i)

inline void **scale_column**(*T* a, size_t i)

Public Members

Matrix<*T*> **S**

Matrix<*T*> **A**

Matrix<*T*> **Sinv**

Struct L1Dist

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_metric.hpp`

Inheritance Relationships

Base Type

- `public bats::AbstractMetric< L1Dist >` (*Template Struct AbstractMetric*)

Struct Documentation

struct **L1Dist** : public bats::*AbstractMetric*<*L1Dist*>

Public Functions

```
template<typename T>
inline T dist(const VectorView<T> &x, const VectorView<T> &y) const
```

Struct LightSimplicialComplex::simplex_boundary_iterator

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_complex_light_simplicial_c`

Nested Relationships

This struct is a nested type of *Template Class LightSimplicialComplex*.

Struct Documentation

struct **simplex_boundary_iterator**

Public Functions

```
inline simplex_boundary_iterator(index_type s, size_t dim, const LightSimplicialComplex *p)

inline simplex_boundary_iterator(index_type s, size_t dim, const LightSimplicialComplex *p, int i)

inline simplex_boundary_iterator(int i)

inline std::tuple<index_type, int> next()

inline index_type operator*() const

inline simplex_boundary_iterator &operator++()

inline simplex_boundary_iterator &operator--()

inline bool operator!=(const simplex_boundary_iterator &other)

inline bool operator==(const simplex_boundary_iterator &other)

inline operator bool() const
```

Public Members

```
const LightSimplicialComplex *p

size_t dim

int c

int i

index_type before

index_type after
```

Struct LInfDist

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_metric.hpp`

Inheritance Relationships

Base Type

- `public bats::AbstractMetric< LInfDist > (Template Struct AbstractMetric)`

Struct Documentation

struct **LInfDist** : public bats::*AbstractMetric*<*LInfDist*>

Public Functions

```
template<typename T>  
inline T dist(const VectorView<T> &x, const VectorView<T> &y) const
```

Struct no_apparent_pairs_flag

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Struct Documentation

struct **no_apparent_pairs_flag**

Struct no_optimization_flag

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Struct Documentation

struct **no_optimization_flag**

Template Struct PersistencePair

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_persistence_barcode.hpp`

Struct Documentation

```
template<typename T>
struct PersistencePair
```

Public Functions

```
inline PersistencePair()
```

```
inline PersistencePair(const size_t dim, const size_t birth_ind, const size_t death_ind, const T birth, const T death)
```

```
inline bool operator==(const PersistencePair &other) const
```

```
inline bool operator!=(const PersistencePair &other) const
```

```
inline size_t get_dim() const
```

```
inline size_t get_birth_ind() const
```

```
inline size_t get_death_ind() const
```

```
inline T get_birth() const
```

```
inline T get_death() const
```

```
inline T length() const
```

```
inline T mid() const
```

```
inline std::string str()
```

Public Members

```
size_t dim
```

```
size_t birth_ind
```

```
size_t death_ind
```

```
T birth
```

```
T death
```

Template Struct ReducedDGVectorSpace

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_dgbasis.hpp`

Struct Documentation

```
template<typename MT>
```

```
struct ReducedDGVectorSpace
```

Public Types

```
using vect_type = typename MT::col_type
```

Public Functions

```
inline size_t hdim(size_t k) const
```

```
inline size_t betti(size_t k) const
```

```
inline size_t maxdim() const
```

```
inline size_t dim(size_t k) const
```

```
inline MT &operator[] (size_t k)
```

```
inline void initialize(const DGVectorSpace<MT> &C)  
    initialize with chain complex C, but do not do reduction
```

```
inline void set_indices()
```

```
inline ReducedDGVectorSpace()
```

```
inline ReducedDGVectorSpace(const DGVectorSpace<MT> &C)
```

```
template<typename algflag>  
inline ReducedDGVectorSpace(const DGVectorSpace<MT> &C, algflag)
```

```
template<typename algflag>  
inline ReducedDGVectorSpace(const DGVectorSpace<MT> &C, algflag, bats::compute_basis_flag)
```

```
template<typename algflag>  
inline ReducedDGVectorSpace(const DGVectorSpace<MT> &C, algflag, bats::clearing_flag)
```

```
template<typename algflag>
```

```

inline ReducedDGVectorSpace(const DGVectorSpace<MT> &C, algflag, bats::clearing_flag,
                             bats::compute_basis_flag)

template<typename algflag>
inline ReducedDGVectorSpace(const DGVectorSpace<MT> &C, algflag, bats::compression_flag)

template<typename TV>
inline TV to_hom_basis(const TV &v, size_t k) const
    put vector/matrix in homology-revealing basis in dimension k

template<typename TV>
inline TV from_hom_basis(const TV &v, size_t k) const
    put vector/matrix back in original basis in dimension k

inline vect_type get_preferred_representative(const size_t j, const size_t k) const

inline void find_preferred_representative(vect_type &y, size_t k) const

inline vect_type chain_preferred_representative(const vect_type &c, size_t k) const

inline void print_summary(bool print_nnz = false) const

inline void _make_U_upper_triangular(size_t k)
    helper function for updates

    Reduces U, permutes to be upper triangular applies same column operations to R

    WARNING: after this, p2c[k] will contain pivots of U

template<typename ...Args>
inline void update_reduction2(size_t k, Args... args)

inline void permute_matrices(size_t k, const std::vector<size_t> &perm)

template<typename ...Args>
inline void permute_basis(const std::vector<std::vector<size_t>> &perm, Args... args)

inline size_t find_reverse_index(const size_t &i, const size_t &n)

inline std::vector<size_t> find_reverse_index(const std::vector<size_t> &index_list, const size_t &n)

inline std::vector<std::vector<size_t>> find_reverse_index(const std::vector<std::vector<size_t>>
                                                         &index_list, const size_t &n)

inline void _delete_cells(size_t k, const UpdateInfo2 &UI)
    delete cells in dimension k

inline void _insert_cells(size_t k, const UpdateInfo2 &UI)

```



```

inline void update_basis(const UpdateInfo2 &UI)
    General update factorization

template<typename Information_type, typename ...Args>
inline void update_basis_general(const Information_type &UI, Args... args)

template<typename Information_type, typename ...Args>
inline void update_basis_general_clearing(const Information_type &UI, Args... args)

```

Public Members

```

int degree

std::vector<MT> U

std::vector<MT> R

std::vector<std::vector<size_t>> I

std::vector<std::vector<size_t>> p2c

```

Template Struct ReducedFilteredChainComplex

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_persistence_filtered_basis.h`

Struct Documentation

```

template<typename T, typename MT>
struct ReducedFilteredChainComplex

```

Public Functions

```

inline ReducedFilteredChainComplex()

template<typename ...Args>
inline ReducedFilteredChainComplex(const FilteredChainComplex<T, MT> &C, Args... args)

inline size_t maxdim() const

inline size_t dim(const size_t k) const

inline size_t hdim(const size_t k) const

inline std::vector<PersistencePair<T>> persistence_pairs(const size_t k, const bool permuted = false)
    const
    persistence pairs in dimension k

```

Parameters

- **k** – homology dimension
- **permuted** – set to true to return critical indices permuted by filtration parameter set to false to return with indices in original order. Default: false

inline std::tuple<std::vector<*T*>, std::vector<size_t>> **persistence_pairs_vec**(const size_t k, const bool permuted = false) const

return persistence pairs in vector format

returns flattened vectors bd - birth-death pairs inds - critical indices

inline auto **representative**(const *PersistencePair*<*T*> &p, const bool permuted = false) const
returns representative for homology class corresponding to persistence pair

if permuted is false, it is assumed the birth index is also not in permutation order.

Parameters

- **p** – persistence pair obtained from persistence_pairs
- **permuted** – set to true to return indices permuted by filtration parameter set to false to return with indices in original order. Default: false

std::vector<*T*> **barcode**(const size_t k)

std::vector<size_t> **critical_cells**(const size_t k)

inline void **update_filtration**(const std::vector<std::vector<*T*>> newval)

template<typename **Information_type**, typename ...**Args**>

inline void **update_filtration_general**(const *Information_type* &updating_information, *Args*... args)

inline std::vector<size_t> **get_nnz_U**()

inline std::vector<size_t> **get_nnz_R**()

inline void **sparsify_basis**()

greedily introduce sparsity into basis

inline void **remove_extra_cycles**()

remove extra cycles from $U[k]$

ReducedFilteredChainComplex **get_subcomplex**() const

inline void **print_summary**(bool print_nnz = false) const

Public Members

ReducedChainComplex<MT> RC

std::vector<std::vector<T>> val

std::vector<std::vector<size_t>> perm

Template Struct ReducedFilteredDGVectorSpace

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_persistence_filtered_dgbasis

Struct Documentation

template<typename T, typename MT>

struct **ReducedFilteredDGVectorSpace**

Public Functions

inline **ReducedFilteredDGVectorSpace**()

template<typename ...Args>

inline **ReducedFilteredDGVectorSpace**(const *FilteredDGVectorSpace*<T, MT> &C, Args... args)

inline size_t **maxdim**() const

inline size_t **dim**(const size_t k) const

inline size_t **hdim**(const size_t k) const

inline std::vector<*PersistencePair*<T>> **persistence_pairs**(const size_t k) const
persistence pairs in dimension k

Parameters k – homology dimension

inline std::tuple<std::vector<T>, std::vector<size_t>> **persistence_pairs_vec**(const size_t k, const bool
permuted = false) const

return persistence pairs in vector format

returns flattened vectors bd - birth-death pairs inds - critical indices

inline void **update_filtration**(const std::vector<std::vector<T>> newval)

inline void **update_basis**(*UpdateInfo2* &UI)

template<typename **Information_type**, typename ...Args>

inline void **update_filtration_general**(const *Information_type* &updating_information, Args... args)

template<typename **Information_type**, typename ...Args>

```
inline void update_filtration_general_clearing(const Information_type &updating_information,  
                                              Args... args)
```

Public Members

```
ReducedDGVectorSpace<MT> RC  
std::vector<std::vector<T>> val  
std::vector<std::vector<size_t>> perm
```

Struct **RPAngleDist**

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_metric.hpp`

Inheritance Relationships

Base Type

- `public bats::AbstractMetric< RPAngleDist > (Template Struct AbstractMetric)`

Struct Documentation

```
struct RPAngleDist : public bats::AbstractMetric<RPAngleDist>
```

Public Functions

```
template<typename T>  
inline T dist(const VectorView<T> &x, const VectorView<T> &y) const
```

Struct **RPCosineDist**

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_metric.hpp`

Inheritance Relationships

Base Type

- `public bats::AbstractMetric< CosineDist > (Template Struct AbstractMetric)`

Struct Documentation

struct **RPCosineDist** : public bats::AbstractMetric<CosineDist>

Public Functions

```
template<typename T>  
inline T dist(const VectorView<T> &x, const VectorView<T> &y) const
```

Struct sparse_reduction_flag

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp

Struct Documentation

struct **sparse_reduction_flag**

Struct standard_reduction_flag

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp

Struct Documentation

struct **standard_reduction_flag**

Template Struct tedge

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_rips.hpp

Struct Documentation

```
template<typename TF, typename TI>  
struct tedge
```

Public Functions

inline **tedge**()

inline **tedge**(*TF* v, *TI* s, *TI* t)

Public Members

TF v

TI s

TI t

Struct triangle

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_extras.hpp`

Struct Documentation

struct **triangle**

Public Functions

inline **triangle**()

inline **triangle**(size_t a, size_t b, size_t c)

Public Members

size_t a

size_t b

size_t c

size_t ab

size_t bc

size_t ca

Template Struct Update_info

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_update_informati`

Struct Documentation

template<class **FiltrationType**>

struct **Update_info**

Struct to hold information for updating *ReducedChainComplex* or *ReducedDGVectorSpace*

FIELDS: maxdim: maximum dimension of cells in a filtration deletion_indices: permutations: addition_indices: boundary_indices: F_Y_vals: F_Y_perms:

ASSUMPTIONS: two filtration dimensions are assumed to be the same!!

Public Functions

inline void **filtered_info**(const std::vector<std::vector<size_t>> &perms_X, const int °ree = -1)

This function is only called in tests

inline **Update_info**(const *FiltrationType* &F_X, const *FiltrationType* &F_Y, int degree = -1)

inline std::vector<size_t> **permutation_deletion_end**(size_t i)

inline void **print_summary**()

inline void **print_detail**()

Public Members

std::vector<std::vector<size_t>> **addition_indices**

std::vector<std::vector<std::vector<size_t>>> **boundary_indices**

std::vector<std::vector<size_t>> **deletion_indices**

std::vector<std::vector<size_t>> **permutations**

std::vector<std::vector<size_t>> **intersection_indices_Y**

std::vector<std::vector<size_t>> **intersection_indices_X**

std::vector<size_t> **kendall_tau_dists**

size_t **max_dim**

std::vector<std::vector<double>> **F_X_vals**

std::vector<std::vector<size_t>> **F_X_perms**

std::vector<std::vector<size_t>> **perms_X_inv**

std::vector<std::vector<double>> **F_Y_vals**

std::vector<std::vector<size_t>> **F_Y_perms**

```
std::vector<std::vector<size_t>> perms_Y_inv
FiltrationType F_old
FiltrationType F_new
bool filtered_boolean = false
int degree = -1
```

Struct UpdateInfo2

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_update_informati`

Struct Documentation

struct **UpdateInfo2**

Struct to hold information to update RU decomposition of one filtration into the RU decomposition of another filtration.

Brad's version

Public Functions

```
inline const std::vector<std::vector<size_t>> &newperm() const
    new permutation for index
```

```
inline void reverse_for_cohomology()
    reverse everything for cohomology
```

```
template<class FiltrationType>
```

```
inline UpdateInfo2(const FiltrationType &F1, const FiltrationType &F2)
```

Compute update information to turn RU decomposition for F1 into RU decomposition for F2

ASSUME: F1.maxdim() == F2.maxdim()

Public Members

```
std::vector<size_t> ndeletions
```

```
std::vector<std::vector<size_t>> insertion_indices
```

```
std::vector<std::vector<SparseVector<int>>> insertion_cols
```

```
std::vector<std::vector<size_t>> perm
```

```
std::vector<std::vector<double>> newval
```

```
std::vector<std::vector<size_t>> perms2
```

```
bool reversed = false
```


Struct SimplexHasher

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_simplex.hpp`

Struct Documentation

struct **SimplexHasher**

Public Functions

inline `std::size_t operator()` (const `std::vector<size_t> &k`) const

Template Struct rfilt_val

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_reduction.hpp`

Struct Documentation

template<typename **T**>

struct **rfilt_val**

A struct that packages information about entries and exits in a right filtration

Public Functions

inline **rfilt_val**()

inline **rfilt_val**(size_t dim, size_t ind, size_t cind, *T* val, bool entry)

Public Members

size_t **dim**

size_t **ind**

size_t **cind**

T **val**

bool **entry**

Friends

```
inline friend std::ostream &operator<<(std::ostream &os, const rfilt_val &v)
```

Template Struct ZigzagChainComplex

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_zigzag_filtration.h`

Struct Documentation

```
template<typename MT, typename T = double>
```

```
struct ZigzagChainComplex
```

A class that wraps a chain complex with a zigzag filtration.

A class that wraps a chain complex with a zigzag filtration. Unlike a Zigzag filtration, every chain has a unique entry and removal time.

Public Functions

```
inline void _correct_indices(size_t k, size_t j, const std::vector<std::vector<size_t>> extra_cells)
```

correct the indices in column j in dimension k

assumes column j hasn't already been corrected assumes `val[k][j]` has been set, as well as `val[k-1]` extra_cells maps to duplicate cells

```
inline ZigzagChainComplex()
```

```
template<typename CpxT>
```

```
inline ZigzagChainComplex(const ZigzagFiltration<CpxT, T> &X)
```

Construct a zigzag chain complex from a zigzag filtration

constructs a distinct column for every time a cell enters

```
inline size_t maxdim() const
```

return maximum dimension of cells

```
inline size_t dim(const size_t k) const
```

return number of cells in specified dimension

Parameters `dim` – dimension

```
inline size_t dim() const
```

```
inline const std::vector<std::vector<std::pair<T, T>>> &vals() const
```

return const reference to right filtration values

```
inline const std::vector<std::pair<T, T>> &vals(const size_t k) const
```

return const reference to right filtration values

Parameters `k` – dimension of values to return

Public Members

ChainComplex<*MT*> **C**

std::vector<std::vector<std::pair<*T*, *T*>>> **val**

std::vector<std::vector<size_t>> **cind**

Template Struct ZigzagPair

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_reduction.hpp

Struct Documentation

template<typename **T**>

struct **ZigzagPair**

Public Functions

inline **ZigzagPair**()

inline **ZigzagPair**(const size_t dim, const size_t birth_ind, const size_t death_ind, const *T* birth, const *T* death, const bool birth_is_entry, const bool death_is_entry)

inline bool **operator==**(const *ZigzagPair* &other) const

inline bool **operator!=**(const *ZigzagPair* &other) const

inline size_t **get_dim**() const

inline size_t **get_birth_ind**() const

inline size_t **get_death_ind**() const

inline *T* **get_birth**() const

inline *T* **get_death**() const

inline *T* **length**() const

inline *T* **mid**() const

inline std::string **str**()

Public Members

```
size_t dim
size_t birth_ind
size_t death_ind
T birth
T death
bool birth_is_entry
bool death_is_entry
```

Friends

```
inline friend std::ostream &operator<<(std::ostream &os, const ZigzagPair &p)
```

Struct `cell_ind`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_complex_abstract_complex`.

Struct Documentation

```
struct cell_ind
```

Public Functions

```
inline cell_ind()

inline cell_ind(size_t dim, size_t ind)

inline std::string str()
```

Public Members

```
size_t dim
size_t ind
```

Template Struct D

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_matrix_interface.hpp`

Inheritance Relationships

Base Types

- `public E< Impl > (Template Struct E)`
- `public L< Impl > (Template Struct L)`
- `public U< Impl > (Template Struct U)`

Struct Documentation

`template<typename Impl>`

`struct D : public E<Impl>, public L<Impl>, public U<Impl>`

Template Struct E

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_matrix_interface.hpp`

Inheritance Relationships

Base Type

- `public A< Impl > (Template Struct A)`

Derived Types

- `public D< Impl > (Template Struct D)`
- `public EL< Impl > (Template Struct EL)`
- `public ELH< Impl > (Template Struct ELH)`
- `public EU< Impl > (Template Struct EU)`
- `public EUH< Impl > (Template Struct EUH)`
- `public P< Impl > (Template Struct P)`

Struct Documentation

template<typename **Impl**>

struct **E** : public *A<Impl>*

Subclassed by *D<Impl>*, *EL<Impl>*, *ELH<Impl>*, *EU<Impl>*, *EUH<Impl>*, *P<Impl>*

Template Struct EL

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_matrix_interface.hpp`

Inheritance Relationships

Base Types

- public *E< Impl >* (*Template Struct E*)
- public *L< Impl >* (*Template Struct L*)

Struct Documentation

template<typename **Impl**>

struct **EL** : public *E<Impl>*, public *L<Impl>*

Template Struct ELH

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_matrix_interface.hpp`

Inheritance Relationships

Base Types

- public *E< Impl >* (*Template Struct E*)
- public *L< Impl >* (*Template Struct L*)

Struct Documentation

template<typename **Impl**>

struct **ELH** : public *E<Impl>*, public *L<Impl>*

Template Struct EU

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_matrix_interface.hpp`

Inheritance Relationships

Base Types

- `public E< Impl > (Template Struct E)`
- `public U< Impl > (Template Struct U)`

Struct Documentation

```
template<typename Impl>
struct EU : public E<Impl>, public U<Impl>
```

Template Struct EUH

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_matrix_interface.hpp`

Inheritance Relationships

Base Types

- `public E< Impl > (Template Struct E)`
- `public U< Impl > (Template Struct U)`

Struct Documentation

```
template<typename Impl>
struct EUH : public E<Impl>, public U<Impl>
```

Template Struct `is_UnivariatePolynomial`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_polynomial.hpp`

Inheritance Relationships

Base Type

- `public false_type`

Struct Documentation

`template<typename T>`

`struct is_UnivariatePolynomial : public false_type`

Template Struct `is_UnivariatePolynomial< UnivariatePolynomial< T > >`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_polynomial.hpp`

Inheritance Relationships

Base Type

- `public true_type`

Struct Documentation

`template<typename T>`

`struct is_UnivariatePolynomial<UnivariatePolynomial<T>> : public true_type`

Template Struct `L`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_matrix_interface.hpp`

Inheritance Relationships

Base Type

- `public T< Impl > (Template Struct T)`

Derived Types

- `public D< Impl > (Template Struct D)`
- `public EL< Impl > (Template Struct EL)`
- `public ELH< Impl > (Template Struct ELH)`

Struct Documentation

template<typename **Impl**>

struct **L** : public *T<Impl>*

Subclassed by *D< Impl >*, *EL< Impl >*, *ELH< Impl >*

Struct MAT

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_matrix_interface.hpp`

Inheritance Relationships

Derived Types

- `public A< Impl > (Template Struct A)`
- `public A< T > (Template Struct A)`

Struct Documentation

struct **MAT**

Subclassed by *A< Impl >*, *A< T >*

Struct MultiGraph::Edge

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_multigraph_multigraph.hpp`

Nested Relationships

This struct is a nested type of *Template Class MultiGraph*.

Struct Documentation

struct **Edge**

Public Functions

inline **Edge**(*Node* &sin, *Node* &tin, ET &x)

inline ET ***get_data**()

inline void **print**()

Public Members

ET ***data**

Node ***src**

Node ***targ**

Struct MultiGraph::Node

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_multigraph_multigraph.hpp`

Nested Relationships

This struct is a nested type of *Template Class MultiGraph*.

Struct Documentation

struct **Node**

Public Functions

inline **Node**(NT &x)

inline **Node**(NT *x)

inline void **add_input**(*Edge* *in)

inline void **add_output**(*Edge* *out)

```
inline NT *get_data()
```

```
inline void print()
```

Public Members

```
NT *data
```

```
std::vector<Edge*> input
```

```
std::vector<Edge*> output
```

Template Struct nzpair

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_abstract_vector.hpp`

Struct Documentation

```
template<typename TI, typename TV>
```

```
struct nzpair
```

Public Functions

```
inline nzpair()
```

```
inline nzpair(const TI ind)
```

```
inline nzpair(const TI ind, const TV val)
```

```
inline nzpair(const nzpair &p)
```

```
inline nzpair(std::string &str)
```

```
inline bool operator==(const nzpair &other) const
```

```
inline bool operator!=(const nzpair &other) const
```

Public Members

TI **ind**

TV **val**

Template Struct P

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_matrix_interface.hpp`

Inheritance Relationships

Base Type

- `public E< Impl > (Template Struct E)`

Struct Documentation

```
template<typename Impl>
```

```
struct P : public E<Impl>
```

Struct SI

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation.hpp`

Struct Documentation

```
struct SI
```

Template Struct SmithFact

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_pid.hpp`

Struct Documentation

```
template<class TC>
```

```
struct SmithFact
```

Public Functions

inline *ColumnMatrix*<*TC*> **prod()** const

Public Members

ColumnMatrix<*TC*> **R**

ColumnMatrix<*TC*> **S**

ColumnMatrix<*TC*> **C**

Template Struct SparseFact

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Struct Documentation

template<class **TC**>

struct **SparseFact**

Public Functions

inline *ColumnMatrix*<*TC*> **LEUP_prod()** const

inline *ColumnMatrix*<*TC*> **PLEU_prod()** const

inline *ColumnMatrix*<*TC*> **UELP_prod()** const

inline *ColumnMatrix*<*TC*> **PUEL_prod()** const

inline *ColumnMatrix*<*TC*> **LQU_prod()** const

inline *ColumnMatrix*<*TC*> **UQL_prod()** const

Public Members

ColumnMatrix<*TC*> **L**

ColumnMatrix<*TC*> **E**

ColumnMatrix<*TC*> **U**

ColumnMatrix<*TC*> **P**

Template Struct T

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_matrix_interface.hpp`

Inheritance Relationships

Base Type

- `public A< Impl > (Template Struct A)`

Derived Types

- `public L< Impl > (Template Struct L)`
- `public U< Impl > (Template Struct U)`

Struct Documentation

template<typename **Impl**>

struct **T** : public *A<Impl>*

Subclassed by *L< Impl >*, *U< Impl >*

Template Struct U

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_matrix_interface.hpp`

Inheritance Relationships

Base Type

- `public T< Impl > (Template Struct T)`

Derived Types

- `public D< Impl > (Template Struct D)`
- `public EU< Impl > (Template Struct EU)`
- `public EUH< Impl > (Template Struct EUH)`

Struct Documentation

```
template<typename Impl>
struct U : public T<Impl>
    Subclassed by D<Impl> , EU<Impl> , EUH<Impl>
```

Template Class AbstractField

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_field.hpp`

Class Documentation

```
template<class Derived>
class AbstractField
```

Public Functions

```
inline Derived operator+(const Derived &b)
```

```
inline Derived operator-(const Derived &b)
```

```
inline Derived operator-()
```

```
inline Derived operator*(const Derived &b)
```

```
inline Derived operator/(const Derived &b)
```

```
inline Derived inv()
```

```
inline Derived operator==(const Derived &b)
```

```
inline Derived operator==(const int b)
```

```
inline void print()
```

```
inline std::string str()
```

Friends

```
inline friend std::ostream &operator<<(std::ostream &os, AbstractField &x)
```

Class AbstractMatrix

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_abstract_matrix.hpp`

Inheritance Relationships

Derived Types

- `public ColumnMatrix< TC >` (*Template Class ColumnMatrix*)
- `public CSCMatrix< TV, TI >` (*Template Class CSCMatrix*)

Class Documentation

```
class AbstractMatrix  
    Subclassed by ColumnMatrix< TC >, CSCMatrix< TV, TI >
```

Template Class BarcodePair

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_barcode.hpp`

Class Documentation

```
template<typename T>  
class BarcodePair
```

Public Functions

```
inline BarcodePair()  
  
inline BarcodePair(T b)  
  
inline BarcodePair(T b, T d)  
  
inline void print()
```


Public Members

T **birth**

T **death**

Class CellComplex

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_complex_cell_complex.hpp`

Class Documentation

class **CellComplex**

Public Functions

inline **CellComplex**()

inline **CellComplex**(size_t maxdim)

inline size_t **maxdim**() const

inline size_t **ncells**(size_t k) const

inline size_t **ncells**() const

inline size_t **add**(const std::vector<size_t> &b, const std::vector<int> &c, size_t k)

inline size_t **add_vertex**()

inline size_t **add_vertices**(size_t k)

inline *CSCMatrix*<int, size_t> **boundary_csc**(size_t dim) const

inline *ColumnMatrix*<*SparseVector*<int, size_t>> **boundary**(size_t dim)

void **boundary**(size_t k, std::vector<size_t> row, std::vector<size_t> col)

Class CellularMap

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_complex_cell_map.hpp`

Class Documentation

class **CellularMap**

Public Functions

inline **CellularMap**()

inline **CellularMap**(size_t dim)

inline **CellularMap**(std::string &fname)

template<typename **CpxT**>

inline **CellularMap**(const *CpxT* &X, const *CpxT* &Y)

Construct inclusion map

Should work for both *SimplicialComplex* and *CubicalComplex*

inline size_t **maxdim**() const

inline map_type &**operator**[](size_t k)

inline const map_type &**operator**[](size_t k) const

inline void **save**(std::string &fname) const

Public Static Functions

template<typename **CpxT**>

static inline *CellularMap* **identity**(const *CpxT* &X)

Class CubicalComplex

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_complex_cubical_complex.h`

Class Documentation

class **CubicalComplex**

Public Functions

inline **CubicalComplex**()

inline **CubicalComplex**(size_t maxdim)

inline **CubicalComplex**(size_t n, size_t maxdim)

inline size_t **find_idx**(const std::vector<size_t> &s)

inline size_t **find_idx**(const std::vector<size_t> &s) const

inline size_t **maxdim**() const

inline size_t **ncells**(const size_t k) const

inline size_t **ncells**() const

inline void **set_dimension**(size_t maxdim)

inline void **print_summary**() const

inline *cell_ind* **add**(std::vector<size_t> &s)

inline *cell_ind* **add**(std::vector<size_t> &&s)

inline std::vector<*cell_ind*> **add_recursive**(const std::vector<size_t> &s)

inline std::vector<*cell_ind*> **add_recursive**(const std::vector<size_t> &&s)

inline auto **faces_begin**(const size_t dim, const size_t i) const

inline auto **faces_begin**(const *cell_ind* &ci) const

inline auto **faces_end**(const size_t dim, const size_t i) const

inline auto **faces_end**(const *cell_ind* &ci) const

```
inline auto cell_begin(const size_t dim, const size_t i) const

inline auto cell_end(const size_t dim, const size_t i) const

inline void get_cube(size_t dim, size_t i, std::vector<size_t> &s) const

inline std::vector<size_t> get_cube(size_t dim, size_t i) const

inline auto get_cell(size_t dim, size_t i, std::vector<size_t> &s) const

inline auto get_cell(size_t dim, size_t i) const

inline std::vector<std::vector<size_t>> get_cubes(const size_t dim) const

inline CubicalComplex skeleton(const size_t k) const

inline std::vector<std::vector<size_t>> get_cubes() const

inline CSCMatrix<int, size_t> boundary_csc(const size_t dim) const

inline void load_cubes(std::string &fname)
```

Public Static Functions

```
static inline CubicalComplex generate_cube(const size_t n)
    generate a discretized cube on  $n^3$  vertices
```

Parameters **n** – number of vertex locations along each dimension

Template Class Diagram

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_multigraph_diagram.hpp`

Nested Relationships

Nested Types

- *Struct Diagram::Edge*

Class Documentation

template<typename **NT**, typename **ET**>

class **Diagram**

Public Functions

inline **Diagram**()

inline **Diagram**(size_t n, size_t m)

inline size_t **nnode**() const

inline size_t **nedge**() const

inline *NT* &**node_data**(size_t i)

inline const *NT* &**node_data**(size_t i) const

inline *ET* &**edge_data**(size_t j)

inline const *ET* &**edge_data**(size_t j) const

inline size_t **edge_source**(size_t j) const

inline size_t **edge_target**(size_t j) const

inline size_t **add_node**(*NT* &a)

inline size_t **add_node**(*NT* &&a)

inline void **set_node**(size_t i, *NT* &a)

inline void **set_node**(size_t i, *NT* &&a)

inline size_t **add_edge**(size_t i, size_t j, *ET* &data)

inline size_t **add_edge**(size_t i, size_t j, *ET* &&data)

inline void **set_edge**(size_t i, size_t s, size_t t, const *ET* &data)

inline void **save_metadata**(std::string &fname) const

```
inline void save(std::string &dname) const
```

Public Members

```
std::vector<NT> node  
std::vector<ET> edata  
std::vector<Edge> elist  
struct Edge
```

Public Functions

```
inline Edge()  
  
inline Edge(size_t s, size_t t)
```

Public Members

```
size_t src  
size_t targ
```

Template Class Filtration

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_filtration.hpp`

Class Documentation

```
template<typename TF, class CpxT>
```

```
class Filtration
```

A filtration which can be used to wrap a simplicial/cubical/cell complex.

A filtration class, templated over the type of the filtration parameter, and the type of the underlying complex

Public Functions

```
inline Filtration()  
  
inline Filtration(const CpxT &C)
```

```
template<class ...Ts>  
inline Filtration(const Ts (&... args))  
    Initialization which passes arguments to initialize the underlying complex
```

Parameters **args...** – passed to complex initialization

```
inline Filtration(const CpxT &C, const std::vector<std::vector<TF>> &vals)
    Initialization which passes arguments to initialize the underlying complex
```

Parameters

- **C** – complex
- **vals** – filtration values for every cell in C

```
inline void update_filtration(const std::vector<std::vector<TF>> &vals2)
    update filtration values
```

Parameters **vals2** – new filtration values

```
inline const CpxT &complex() const
    return const reference to underlying complex
```

```
inline const std::vector<std::vector<TF>> &vals() const
    return const reference to filtration values
```

```
inline const std::vector<TF> &vals(const size_t k) const
    return const reference to filtration values
```

Parameters **k** – dimension of values to return

```
inline size_t maxdim() const
    return maximum dimension of cells
```

```
inline size_t ncells(const size_t dim) const
    return number of cells in specified dimension
```

Parameters **dim** – dimension

```
template<class ...Ts>
inline cell_ind add(TF t, Ts (&... args))
    add cell to filtration
```

Parameters

- **t** – filtration parameter
- **...args** – passed to add method of underlying complex

```
template<class ...Ts>
inline cell_ind add_directed(TF t, Ts (&... args))
    add directed cell to filtration
```

Parameters

- **t** – filtration parameter
- **...args** – passed to add method of underlying complex

```
template<class ...Ts>
inline std::vector<cell_ind> add_recursive(TF t, Ts (&... args))
    Add recursively to filtration. Any cells added will take filtration value t.
```

Parameters

- **t** – filtration parameter
- **...args** – passed to add_recursive method of underlying complex

```
inline void union_add(const Filtration &F2)
```

```
inline std::vector<size_t> sortperm(size_t dim) const
```

```
inline std::vector<std::vector<size_t>> sortperm() const
```

```
inline CpxT sublevelset(const TF a) const
```

Get sub-levelset of filtration $(-\infty, a]$

Parameters *a* – upper bound of levelset

Class CompositePermutation

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_permutation.hpp`

Class Documentation

```
class CompositePermutation
```

Public Functions

```
inline CompositePermutation()
```

```
inline void append(const ElementaryPermutation &p)
```

```
inline void swap(size_t i, size_t j)
```

```
template<typename T>
```

```
inline T &operator()(T &a) const
```

```
template<typename T>
```

```
inline T &operator()(T &&a) const
```

```
inline CompositePermutation &operator()(const ElementaryPermutation &p)
```

Friends

```
friend std::ostream &operator<<(std::ostream &os, const CompositePermutation &p)
```


Template Class `const_strided_iterator`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_dense.hpp`

Inheritance Relationships

Base Type

- `public std::iterator< std::random_access_iterator_tag, T >`

Class Documentation

template<typename T>

class **const_strided_iterator** : public std::iterator<std::random_access_iterator_tag, *T*>

Public Functions

inline **const_strided_iterator**()

inline **const_strided_iterator**(const *T* *v, ssize_t s)

inline ~**const_strided_iterator**()

inline iterator **operator++**(int)

inline iterator &**operator++**()

inline iterator **operator--**(int)

inline iterator &**operator--**()

inline reference **operator***() const

inline pointer **operator-->**() const

inline bool **operator==(const iterator &rhs)** const

inline bool **operator!=(const iterator &rhs)** const

inline iterator **operator+**(size_t i) const

inline reference **operator[]**(size_t i) const

Template Class Matrix

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_dense.hpp`

Class Documentation

template<typename **T**>

class **Matrix**

Public Types

typedef *T* **value_type**

Public Functions

inline size_t **len**() const

inline void **fill**(*T* a)

inline **Matrix**()

inline **Matrix**(size_t m, size_t n, *RowMajor*)

inline **Matrix**(size_t m, size_t n, *ColumnMajor*)

inline **Matrix**(size_t m, size_t n)

inline **Matrix**(size_t m, size_t n, *T* a)

template<typename **Major**>

inline **Matrix**(size_t m, size_t n, *T* a, *Major* &ms)

inline **Matrix**(*Matrix* &other, *ColumnMajor*)

inline *T* ***data**()

inline const *T* ***data**() const

inline size_t **nrow**() const

inline size_t **ncol**() const

```
inline std::pair<size_t, size_t> dims() const

inline T operator[](size_t k) const

inline T &operator[](size_t k)

inline T operator()(size_t k) const

inline T &operator()(size_t k)

template<typename I1, typename I2>
inline MatrixView<T, I1, I2> view(I1 &rows, I2 &cols)

template<typename I1, typename I2>
inline MatrixView<T, I1, I2> view(I1 &&rows, I2 &&cols)

template<typename I1, typename I2>
inline const MatrixView<T, I1, I2> view(I1 &rows, I2 &cols) const

template<typename I1, typename I2>
inline const MatrixView<T, I1, I2> view(I1 &&rows, I2 &&cols) const

inline T operator()(size_t i, size_t j) const

inline T &operator()(size_t i, size_t j)

inline VectorView<T> row(size_t i)

inline VectorView<T> column(size_t j)

inline void print_info() const

inline void print() const

inline Matrix transpose() const

inline void swap_rows(size_t i1, size_t i2)

inline void swap_columns(size_t j1, size_t j2)

inline bool append_column(const T val = T(0))

inline bool delete_column()
```

```
inline void add_row(T a, size_t i1, size_t i0)

inline void add_row(size_t i1, size_t i0)

inline void add_column(T a, size_t j1, size_t j0)

inline void add_column(size_t j1, size_t j0)

inline void scale_row(T a, size_t i)

inline void scale_column(T a, size_t j)

template<typename TB>
inline auto mm(const TB &B) const

template<typename TB>
inline Matrix operator*(const TB &B) const

template<typename Tx>
inline auto mv(const Tx &x) const
```

Public Static Functions

```
static inline Matrix identity(size_t n)
```

Template Class MatrixView

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_dense.hpp`

Class Documentation

```
template<typename T, typename I1, typename I2>
class MatrixView
```

Public Types

```
typedef T val_type
```

Public Functions

```
inline MatrixView()
```

```
inline MatrixView(Matrix<T> *m, I1 r, I2 c)
```

```
inline MatrixView(Matrix<T> &m, I1 r, I2 c)
```

```
inline T operator() (size_t i, size_t j) const
```

```
inline T &operator() (size_t i, size_t j)
```

```
inline size_t nrow() const
```

```
inline size_t ncol() const
```

```
inline void print_info() const
```

```
inline void print() const
```

Class Permutation

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_permutation.hpp`

Class Documentation

```
class Permutation
```

Public Functions

```
inline Permutation()
```

```
inline Permutation(size_t n)
```

Template Class `range`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_dense.hpp`

Nested Relationships

Nested Types

- *Class `range::const_iterator`*

Class Documentation

```
template<typename T>
```

```
class range
```

Public Types

```
typedef const_iterator iterator
```

Public Functions

```
inline range()
```

```
inline range(T b, T e)
```

```
inline range(T b, T e, T s)
```

```
inline T first() const
```

```
inline T last() const
```

```
inline T stride() const
```

```
inline size_t size() const
```

```
inline T operator[](size_t i) const
```

```
inline const_iterator const begin()
```

```
inline const_iterator const end()
```

```
inline const_iterator const cbegin()
```

```
inline const_iterator const cend()
```

```
class const_iterator : public std::iterator<std::random_access_iterator_tag, T>
```

Public Functions

```
inline const_iterator()
```

```
inline const_iterator(const T v, const T s)
```

```
inline ~const_iterator()
```

```
inline iterator operator++(int)
```

```
inline iterator &operator++()
```

```
inline iterator operator--(int)
```

```
inline iterator &operator--()
```

```
inline reference operator*() const
```

```
inline pointer operator->() const
```

```
inline bool operator==(const iterator &rhs) const
```

```
inline bool operator!=(const iterator &rhs) const
```

Class `range::const_iterator`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_dense.hpp`

Nested Relationships

This class is a nested type of *Template Class range*.

Inheritance Relationships

Base Type

- `public std::iterator< std::random_access_iterator_tag, T >`

Class Documentation

class **const_iterator** : public std::iterator<std::random_access_iterator_tag, *T*>

Public Functions

inline **const_iterator**()

inline **const_iterator**(const *T* v, const *T* s)

inline ~**const_iterator**()

inline iterator **operator++**(int)

inline iterator &**operator++**()

inline iterator **operator--**(int)

inline iterator &**operator--**()

inline reference **operator***() const

inline pointer **operator-->**() const

inline bool **operator==(**const iterator &rhs) const

inline bool **operator!=(**const iterator &rhs) const

Template Class Span

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_span.hpp`

Class Documentation

template<typename **T**>

class **Span**

Public Functions

inline **Span**()

inline **Span**(*T*)

inline **Span**(size_t n, *T*)

inline size_t **dim**() const

inline size_t **vdim**() const

template<typename **TV**>
inline bool **add**(const *TV* &v)

template<typename **TV**>
inline bool **contains**(const *TV* &v) const

Public Members

Matrix<*T*> **Pt**

Matrix<*T*> **L**

size_t **_vdim**

Template Class **strided_iterator**

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_dense.hpp`

Inheritance Relationships

Base Type

- public `std::iterator< std::random_access_iterator_tag, T >`

Class Documentation

template<typename T>

class **strided_iterator** : public std::iterator<std::random_access_iterator_tag, T>

Public Functions

inline **strided_iterator**()

inline **strided_iterator**(T *v, ssize_t s)

inline ~**strided_iterator**()

inline iterator **operator++**(int)

inline iterator &**operator++**()

inline iterator **operator--**(int)

inline iterator &**operator--**()

inline reference **operator***() const

inline pointer **operator-->**() const

inline bool **operator==**(const iterator &rhs) const

inline bool **operator!=**(const iterator &rhs) const

inline iterator **operator+**(size_t i) const

inline reference **operator[]**(size_t i) const

Template Class VectorView

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_dense.hpp`

Class Documentation

template<typename **T**>

class **VectorView**

Public Types

typedef *strided_iterator*<*T*> **iterator**

typedef *const_strided_iterator*<*T*> **const_iterator**

Public Functions

inline **VectorView**(*T* *s, *T* *e, size_t stride)

inline *iterator* **begin**()

inline *iterator* **end**()

inline *const_iterator* **begin**() const

inline *const_iterator* **end**() const

inline *const_iterator* **cbegin**() const

inline *const_iterator* **cend**() const

inline *T* &**operator**[] (size_t i)

inline const *T* &**operator**[] (size_t i) const

inline void **print**() const

inline void **axpy**(*T* a, const *VectorView* &other)

inline void **axpy**(*T* a, const *VectorView* &&other)

inline void **scale**(*T* a)

Template Class `LightSimplicialComplex`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_complex_light_simplicial_c`

Nested Relationships

Nested Types

- *Struct `LightSimplicialComplex::simplex_boundary_iterator`*

Class Documentation

```
template<typename index_type = size_t, typename hash_table = std::unordered_map<index_type, size_t>>
class LightSimplicialComplex
```

Public Functions

```
inline index_type max_vertex(index_type s, size_t dim) const
```

```
inline LightSimplicialComplex()
```

```
inline LightSimplicialComplex(const index_type n, const index_type k)
```

```
inline index_type maxdim() const
```

```
inline size_t ncells(size_t dim) const
```

```
inline size_t ncells() const
```

```
inline index_type simplex_key(const std::vector<index_type> &s) const
```

```
inline void key_to_simplex(const size_t dim, index_type key, std::vector<index_type> &s) const
```

```
inline std::vector<index_type> key_to_simplex(const size_t dim, index_type key) const
```

```
inline std::vector<index_type> get_simplex(const size_t dim, const size_t i) const
```

```
inline void get_simplex(const size_t dim, const size_t i, std::vector<index_type> &s) const
```

```
inline auto get_cell(size_t dim, size_t i, std::vector<index_type> &s) const
```

```
inline auto get_cell(size_t dim, size_t i) const
```

```
inline std::vector<std::vector<index_type>> get_simplices(const size_t dim) const

inline std::vector<std::vector<size_t>> get_simplices() const

inline size_t find_idx(const size_t dim, const index_type key) const

inline size_t find_idx(const std::vector<index_type> &s) const

inline cell_ind add_unsafe(const size_t dim, const index_type k)

inline cell_ind add(const size_t dim, const index_type k)

inline cell_ind add_unsafe(const std::vector<index_type> &s)

inline auto add(const std::vector<index_type> &s)

inline std::vector<cell_ind> add_recursive(size_t dim, const index_type k)

inline std::vector<cell_ind> add_recursive(const std::vector<index_type> &s)

inline auto simplex_begin(const size_t dim, const size_t i) const

inline auto simplex_end(const size_t dim, const size_t i) const

inline auto boundary(const size_t dim, const size_t i) const

inline CSCMatrix<int, size_t> boundary_csc(const size_t dim) const

inline void print_summary() const
```

Template Class `ReducedChainComplex`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_basis.hpp`

Class Documentation

```
template<typename MT>
```

```
class ReducedChainComplex
```

Public Types

```
using chain_type = typename MT::col_type
```

Public Functions

```
inline size_t hdim(size_t k) const
```

```
inline size_t betti(size_t k) const
```

```
inline size_t maxdim() const
```

```
inline size_t dim(size_t k) const
```

```
inline MT &operator[](size_t k)
```

```
inline void initialize(const ChainComplex<MT> &C)  
    initialize with chain complex C, but do not do reduction
```

```
inline ReducedChainComplex()
```

```
inline ReducedChainComplex(const ChainComplex<MT> &C)
```

```
template<typename algflag>  
inline ReducedChainComplex(const ChainComplex<MT> &C, algflag)
```

```
template<typename algflag>  
inline ReducedChainComplex(const ChainComplex<MT> &C, algflag, bats::compute_basis_flag)
```

```
template<typename algflag>  
inline ReducedChainComplex(const ChainComplex<MT> &C, algflag, bats::clearing_flag)
```

```
template<typename algflag>  
inline ReducedChainComplex(const ChainComplex<MT> &C, algflag, bats::clearing_flag,  
    bats::compute_basis_flag)
```

```
template<typename algflag>  
inline ReducedChainComplex(const ChainComplex<MT> &C, algflag, bats::compression_flag)
```

```
template<typename algflag>  
inline ReducedChainComplex(const ChainComplex<MT> &C, algflag, bats::compression_flag,  
    bats::compute_basis_flag)
```

```
template<typename ...Args>  
inline void update_reduction2(size_t k, Args... args)
```

```

template<typename ...Args>
inline void update_reduction(size_t k, Args... args)

inline void permute_matrices(size_t k, const std::vector<size_t> &perm)

template<typename ...Args>
inline void permute_basis(const std::vector<std::vector<size_t>> &perm, Args... args)

template<typename Information_type, typename ...Args>
inline void update_basis_general(const Information_type &UI, Args... args)

inline void sparsify_basis()
    greedily introduce sparsity into basis

inline void remove_extra_cycles()
    remove extra cycles from  $U[k]$ 

template<typename TV>
inline TV to_hom_basis(const TV &v, size_t k) const

template<typename TV>
inline TV from_hom_basis(const TV &v, size_t k) const

inline chain_type get_preferred_representative(const size_t j, const size_t k) const

inline void find_preferred_representative(chain_type &y, const size_t k) const

inline chain_type chain_preferred_representative(const chain_type &c, size_t k) const

inline void print_summary(bool print_nnz = false) const

```

Public Members

```

std::vector<MT> U
std::vector<MT> R
std::vector<std::vector<size_t>> I
std::vector<std::vector<size_t>> p2c

```

Template Class `ReducedCochainComplex`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_cohom_basis.hp`

Class Documentation

template<typename `MT`>

class `ReducedCochainComplex`

Public Types

using `cochain_type` = typename `MT::col_type`

Public Functions

inline size_t `hdim`(size_t k) const

inline size_t `maxdim`() const

inline `MT` &`operator`[(size_t k)]

inline `ReducedCochainComplex`()

inline `ReducedCochainComplex`(const `CochainComplex`<`MT`> &`C`)

template<typename `algflag`>

inline `ReducedCochainComplex`(const `CochainComplex`<`MT`> &`C`, `algflag`)

template<typename `algflag`>

inline `ReducedCochainComplex`(const `CochainComplex`<`MT`> &`C`, `algflag`, bats::`compute_basis_flag`)

template<typename `algflag`>

inline `ReducedCochainComplex`(const `CochainComplex`<`MT`> &`C`, `algflag`, bats::`clearing_flag`)

template<typename `algflag`>

inline `ReducedCochainComplex`(const `CochainComplex`<`MT`> &`C`, `algflag`, bats::`compression_flag`)

template<typename `TV`>

inline `TV` `to_hom_basis`(const `TV` &`v`, size_t k) const

template<typename `TV`>

inline `TV` `from_hom_basis`(const `TV` &`v`, size_t k) const


```
inline cochain_type get_preferred_representative(size_t i, const size_t k) const
```

```
inline void find_preferred_representative(typename MT::col_type &y, const size_t k) const
```

```
inline cochain_type chain_preferred_representative(const cochain_type &c, size_t k) const
```

Public Members

```
std::vector<size_t> dim
```

```
std::vector<MT> U
```

```
std::vector<MT> R
```

```
std::vector<std::vector<size_t>> I
```

```
std::vector<std::vector<size_t>> p2c
```

Class SimplicialComplex

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_complex_simplicial_complex`

Class Documentation

class **SimplicialComplex**

A simplicial complex based using a trie data structure.

A class which can be used to hold simplicial complexes on large or expanding vertex sets. For a lighter-weight option, see *LightSimplicialComplex*

Public Functions

```
inline SimplicialComplex()
```

```
inline SimplicialComplex(size_t maxdim)
```

Initialization up to a certain dimension

Parameters `maxdim` – [in] - the maximum dimension simplex expected to be added

```
inline SimplicialComplex(size_t n, size_t maxdim)
```

Initialization on a certain vertex

Parameters

- `n` – [in] - the maximum expected vertex index
- `maxdim` – [in] - the maximum dimension simplex expected to be added

```
inline SimplicialComplex(const std::vector<size_t> &dim)
```

```
inline SimplicialComplex(const std::string &&fname)
```

```
inline size_t find_idx(const std::vector<size_t> &s)
    Find the index of a simplex

    Parameters s – [in] A vector containing the simplex

    Returns The index associated with the simplex. Returns bats::NO_IND if the simplex is not in
    the complex.

inline size_t find_idx(const std::vector<size_t> &s) const

inline size_t maxdim() const

inline size_t ncells(const size_t k) const

inline size_t ncells() const

inline cell_ind add(std::vector<size_t> &s)

inline cell_ind add(std::vector<size_t> &&s)

inline cell_ind add_directed(const std::vector<size_t> &s)

inline std::vector<cell_ind> add_recursive(std::vector<size_t> &s)

inline std::vector<cell_ind> add_recursive(std::vector<size_t> &&s)

inline std::vector<cell_ind> add_directed_recursive(std::vector<size_t> &s)

inline auto faces_begin(const size_t dim, const size_t i) const

inline auto faces_begin(const cell_ind &ci) const

inline auto faces_end(const size_t dim, const size_t i) const

inline auto faces_end(const cell_ind &ci) const

inline auto simplex_begin(const size_t dim, const size_t i) const

inline auto simplex_end(const size_t dim, const size_t i) const

inline void get_simplex(size_t dim, size_t i, std::vector<size_t> &s) const
    fill s with simple i in dimension dim

inline std::vector<size_t> get_simplex(size_t dim, size_t i) const
    return simplex i in dimension dim

inline auto get_cell(size_t dim, size_t i, std::vector<size_t> &s) const
```

```

inline auto get_cell(size_t dim, size_t i) const

inline std::vector<std::vector<size_t>> get_simplices(const size_t dim) const

inline std::vector<std::vector<size_t>> get_simplices() const

inline void union_add(const SimplicialComplex &Y)

inline std::vector<size_t> get_indices(const SimplicialComplex &A, size_t dim) const

inline std::vector<std::vector<size_t>> get_indices(const SimplicialComplex &A) const

inline auto boundary(const size_t dim, const size_t k) const
    indices and values of simplex k in dimension dim

inline CSCMatrix<int, size_t> boundary_csc(const size_t dim) const

inline ~SimplicialComplex()

inline void save(std::string &fname) const

inline void print_summary() const

inline void print_cells() const

```

Friends

```

friend class MorsePairing< SimplicialComplex >

inline friend SimplicialComplex simplicial_union(const SimplicialComplex &X, const SimplicialComplex
                                                &Y)

inline friend SimplicialComplex intersection(const SimplicialComplex &X, const SimplicialComplex &Y)

```

Template Class SparseTrie

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_trie.hpp`

Class Documentation

template<typename **A**, typename **T**>

class **SparseTrie**

Public Types

typedef *SparseTrie*<*A*, *T*> **child_type**

typedef std::unordered_map<*A*, *child_type**> **child_container**

Public Functions

inline **SparseTrie**(const *SparseTrie* &t)

inline **SparseTrie**(*SparseTrie* &&t)

inline **SparseTrie**(*T* v)

inline **SparseTrie**()

inline ~**SparseTrie**()

inline *SparseTrie* &**operator**=(const *SparseTrie* &t)

inline *SparseTrie* &**operator**=(*SparseTrie* &&t)

template<typename **ITT**>

inline void **insert**(*ITT* stptr, const *ITT* endptr, const *T* &v)

inline void **emplace**(const std::vector<*A*> &k, *T* &&v)

inline void **emplace**(const std::vector<*A*> &k, const *T* &v)

template<typename **ITT**>

inline *T* &**get**(*ITT* stptr, const *ITT* endptr)

inline *T* &**operator**[] (const std::vector<*T*> &k)

template<typename **ITT**>

inline *T* **get**(*ITT* stptr, const *ITT* endptr, const *T* &def_ret)

inline *T* **get**(const std::vector<*T*> &k, const *T* &def_ret)

```
template<typename ITT>
inline T get(ITT stptr, const ITT endptr, const T &def_ret) const
```

```
inline T get(const std::vector<T> &k, const T &def_ret) const
```

```
template<typename ITT>
inline size_t count(ITT stptr, const ITT endptr)
```

```
inline size_t count(const std::vector<T> &k)
```

Public Members

T **val**

child_container ***children** = nullptr

Class SimplexContainer

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_simplex.hpp`

Class Documentation

class **SimplexContainer**

Public Functions

```
inline SimplexContainer(size_t d)
```

```
inline void emplace_back(std::vector<size_t> &s)
```

```
inline size_t size() const
```

```
inline size_t dim() const
```

Template Class ZigzagFiltration

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_zigzag_filtration.hpp`

Class Documentation

template<typename **CpxT**, typename **T** = double>

class **ZigzagFiltration**

A class that wraps a complex with a zigzag filtration.

A class that wraps a complex with a right filtration. Cells can have entry times and removal times. These intervals are stored as `std::pair<T, T>`

Public Functions

inline **ZigzagFiltration**()

template<class ...**Ts**>

inline **ZigzagFiltration**(const *Ts* (&... args))

Initialization which passes arguments to initialize the underlying complex

Parameters **args...** – passed to complex initialization

inline **ZigzagFiltration**(const *CpxT* &**X**, const std::vector<std::vector<std::vector<std::pair<*T*, *T*>>>> &**val**)

Construct right filtration explicitly on a complex

val should be a vector of vector of pairs. `val[k][i]` is the pair of entry times for cell *i* in dimension *k* in **X**.

No checks are done to make sure the number of values matches the number of cells

Parameters

- **X** – complex representing topological space
- **val** – right filtration values for each cell in **X**

inline const *CpxT* &**complex**() const
return const reference to underlying complex

inline const std::vector<std::vector<std::vector<std::pair<*T*, *T*>>>> &**vals**() const
return const reference to right filtration values

inline const std::vector<std::vector<std::pair<*T*, *T*>>> &**vals**(const size_t **k**) const
return const reference to right filtration values

Parameters **k** – dimension of values to return

inline size_t **maxdim**() const
return maximum dimension of cells

inline size_t **ncells**(const size_t **dim**) const
return number of cells in specified dimension

Parameters **dim** – dimension

template<class ...**Ts**>

inline *cell_ind* **add**(const *T* entry, const *T* exit, *Ts* (&... args))
add cell to right filtration

Parameters

- **entry** – entry parameter

- **exit** – exit parameter
- **...args** – - passed to add method of underlying complex

```
inline cell_ind add(const T entry, const T exit, std::vector<size_t> &&s)
```

```
template<class ...Ts>
```

```
inline std::vector<cell_ind> add_recursive(const T entry, const T exit, Ts (&... args))
```

Add recursively to right filtration. Any cells added will take same entry and exit parameters

Parameters

- **entry** – entry parameter
- **exit** – exit parameter
- **...args** – - passed to add_recursive method of underlying complex

```
inline std::vector<cell_ind> add_recursive(const T entry, const T exit, std::vector<size_t> &&s)
```

```
inline CpxT levelset(T s0, T s1) const
```

Return thickened levelset $f^{-1}([s_0, s_1])$ adds a cell to levelset if $[b, d] [s_0, s_1]$ is non-empty

Parameters

- **s0** – lower bound of levelset
- **s1** – upper bound of levelset

Template Class ColumnMatrix

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_col_matrix.hpp`

Inheritance Relationships

Base Type

- public `AbstractMatrix` (*Class AbstractMatrix*)

Class Documentation

```
template<class TC>
```

```
class ColumnMatrix : public AbstractMatrix
```

Public Types

```
using val_type = typename TC::val_type
using col_type = TC
using tmp_type = typename TC::tmp_type
```

Public Functions

```
inline ColumnMatrix()

inline ColumnMatrix(size_t m, size_t n)

inline ColumnMatrix(size_t _m, size_t _n, val_type a)
    construct column matrix filled with entry a
inline ColumnMatrix(const std::vector<TC> &_col)

inline ColumnMatrix(size_t _m, size_t _n, const std::vector<TC> &_col)

template<typename TC2>
inline ColumnMatrix(const ColumnMatrix<TC2> &other)

inline ColumnMatrix(const CSCMatrix<int, size_t> &A)

inline void read(std::istream &io)

inline ColumnMatrix(std::istream &io)

inline ColumnMatrix(std::string &fname)

inline size_t nrow() const

inline size_t ncol() const

inline std::vector<TC> &cols()

inline const std::vector<TC> &cols() const

inline void set_nrow(size_t mnew)

inline auto getval(const size_t i, const size_t j) const

inline void permute_rows(const std::vector<size_t> &rowperm)
```



```

inline void ipermute_rows(const std::vector<size_t> &rowperm)

template<class ...Ts>
inline void append_column(Ts (&... args))
    append a column to the end of the matrix

    calls a column vector constructor on arguments passed in.

template<class ...Ts>
inline void append_column(Ts (&&... args))

inline void append_row()
    append an empty row to the end of the matrix

template<class ...Ts>
inline void insert_column(const size_t &index, Ts (&&... args))
    Do not recommend, because if we want to insert multiple columns, the index will change after the first
    insertion. Thus the important thing we need to assume is the index list are in ascending order!

inline void append_column()

inline void insert_column(const size_t &index)

inline void insert_columns(const std::vector<size_t> &c_inds, std::vector<TC> &insert_col)
    insert list of columns at list of specified indices

    mutates input inserted columns

inline void append_row(const std::vector<val_type> &row)

inline void append_sparse_row(const col_type &new_row)

inline void insert_row(const size_t &ind, const std::vector<val_type> &row)

inline void insert_sparse_rows(const std::vector<size_t> &r_inds, const std::vector<col_type>
    &new_rows)
    Insert sparse rows by permutation Input: r_inds: new row indices new_rows: a vector of sparse row vectors

inline void insert_rows(const std::vector<size_t> &r_inds, const std::vector<std::vector<size_t>>
    &r_col_inds, const std::vector<std::vector<val_type>>> &r_col_vals)

inline void insert_row(const size_t &ind)

inline void insert_rows(const std::vector<size_t> &r_inds)
    insert zero rows at specified locations

inline void insert_rows(const std::vector<size_t> &r_inds, const std::vector<TC> &rows)
    insert rows represented by sparse vectors

inline void erase_column(const size_t &index)
    insert zero rows at specified locations new_rows[i] at index r_inds[i]

inline void erase_column()

```

```
inline void erase_final_columns(const size_t &ndelete)
    deletes ndelete columns at end of matrix

inline void erase_initial_cols(const size_t ndelete)
    deletes ndelete columns at beginning of matrix

inline void erase_row(const size_t &index)
    erase specified row

inline void erase_row()
    erase last row

inline void erase_rows(size_t n_delete_rows)
    erase last a few rows

inline void erase_row_unsafe()
    assumes that last row is zero so we only need to decrement number of rows

inline void erase_final_rows_unsafe(size_t n_delete_rows)
    assumes that last row is zero so we only need to decrement number of rows

inline void erase_initial_rows(size_t ndelete)
    erases initial rows

inline TC &operator[](size_t index)

inline const TC &operator[](size_t index) const

inline auto operator()(size_t i, size_t j) const

inline bool operator==(const ColumnMatrix &other) const

inline std::vector<std::vector<val_type>> to_row_array() const

inline val_type *dump_dense() const

inline ColumnMatrix submatrix(const std::vector<size_t> &rind, const std::vector<size_t> &cind) const

inline ColumnMatrix block(const size_t i0, const size_t i1, const size_t j0, const size_t j1) const

inline void set_block(const size_t i0, const size_t i1, const size_t j0, const size_t j1, const ColumnMatrix
    &B)

inline void clear_rows(const std::vector<bool> &c)
    clear rows i for which c[i] is true use vector of bools for quick lookup - vector of inds would require search

inline void clear_cols(const std::vector<bool> &c)
    clear columns j for which c[j] is true

    frees memory as well

inline void swap_rows(const size_t i, const size_t i2)
```

```
inline void mix_rows(const size_t i, const size_t i2, const val_type &a, const val_type &b, const val_type &c,
                    const val_type &d)

inline void add_rows(const size_t i, const val_type &c, const size_t i2)

inline TC gemv(const TC &x) const

inline TC gemv(const TC &x, tmp_type &tmpx) const

inline ColumnMatrix operator*(const val_type a) const

inline TC operator*(const TC &x) const

inline ColumnMatrix operator*(const ColumnMatrix &B) const

inline ColumnMatrix operator+(const ColumnMatrix &B) const

inline ColumnMatrix &operator+=(const ColumnMatrix &B)

inline ColumnMatrix operator-(const ColumnMatrix &B) const

inline ColumnMatrix transpose() const

inline ColumnMatrix T() const

inline void permute_cols(const std::vector<size_t> &colperm)

inline void ipermute_cols(const std::vector<size_t> &colperm)

inline void mix_cols(const size_t j, const size_t k, const val_type &a, const val_type &b, const val_type &c,
                    const val_type &d)

inline void permute(const std::vector<size_t> &rowperm, const std::vector<size_t> &colperm)

inline ColumnMatrix &J_right_inplace()

inline ColumnMatrix J_right() const

inline ColumnMatrix &J_left_inplace()

inline ColumnMatrix J_left() const

inline ColumnMatrix &J_conjugation_inplace()
```

```
inline ColumnMatrix J_conjugation() const

inline void swap_cols(size_t j1, size_t j2)

inline void schur_complement(size_t i, size_t j)

inline ColumnMatrix &row_scale(const std::vector<val_type> &coeff)

inline ColumnMatrix &col_inv_scale(const std::vector<val_type> &coeff)

inline size_t nnz() const

inline bool is_zero() const

inline bool is_upper() const

inline bool is_upper_invert() const

inline bool is_lower() const

inline bool is_reduced() const

inline bool is_pivot_matrix() const

inline bool is_EL() const

inline bool is_EU() const

inline bool is_ELhat() const

inline bool is_EUhat() const

inline void print_size() const

inline void print() const

inline void write(std::ostream &io) const

inline std::string str()

inline void save(std::string &fname) const
```

Public Static Functions

static inline *ColumnMatrix* **identity**(size_t n)

static inline *ColumnMatrix* **random**(size_t m, size_t n, double p, int maxval, std::default_random_engine &generator)

static inline *ColumnMatrix* **random**(size_t m, size_t n, double p, int maxval)

Friends

inline friend *TC* **operator***(const *TC* &x, const *ColumnMatrix* &A)
multiplication $x^T A$

inline friend *ColumnMatrix* **tensor_product**(const *ColumnMatrix* &A, const *ColumnMatrix* &B)

inline friend *ColumnMatrix* **direct_sum**(const *ColumnMatrix* &A, const *ColumnMatrix* &B)

Template Class CSCMatrix

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_csc_matrix.hpp`

Inheritance Relationships

Base Type

- public `AbstractMatrix` (*Class AbstractMatrix*)

Class Documentation

template<typename *TV*, typename *TI* = size_t>

class **CSCMatrix** : public *AbstractMatrix*

Public Functions

inline **CSCMatrix**()

inline **CSCMatrix**(size_t m, size_t n, const std::vector<*TI*> &colptr, const std::vector<*TI*> &rowind, const std::vector<*TV*> &val)

inline **CSCMatrix**(const std::vector<*TI*> &colptr, const std::vector<*TI*> &rowind, const std::vector<*TV*> &val)

```
inline TV getval(size_t i, size_t j) const

inline const std::vector<TI> &get_colptr() const

inline const std::vector<TI> &get_rowind() const

inline const std::vector<TV> &get_val() const

inline size_t nrow() const

inline size_t ncol() const

inline void print_size() const

inline void print(size_t rowmin, size_t rowmax, size_t colmin, size_t colmax) const

inline void print() const

inline CSCMatrix submatrix(const std::vector<size_t> &rind, const std::vector<size_t> &cind) const

inline CSCMatrix operator*(const CSCMatrix &other) const
```

Friends

```
inline friend void block_select(const CSCMatrix &M, const std::vector<size_t> &cind, const
                                std::vector<size_t> &prow, const size_t m, CSCMatrix &A)

template<size_t N>
inline friend void block_select(const CSCMatrix &M, const std::vector<size_t> &cind, const
                                std::vector<size_t> *(&&prow)[N], const size_t (&&m)[N], CSCMatrix
                                *(&&A)[N])

inline friend void gemm(const CSCMatrix &A, const CSCMatrix &B, CSCMatrix &C)

inline friend void sum(const CSCMatrix &A, const CSCMatrix &B, CSCMatrix &C)

inline friend void sum(const TV &alpha, const CSCMatrix &A, const CSCMatrix &B, CSCMatrix &C)

inline friend void trilu(const CSCMatrix &A, const CSCMatrix &B, CSCMatrix &C)
```

Template Class ModP

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_field.hpp`

Inheritance Relationships

Base Type

- `public AbstractField< ModP< IntT, P > >` (*Template Class AbstractField*)

Class Documentation

```
template<typename IntT, unsigned P>
```

```
class ModP : public AbstractField<ModP<IntT, P>>
```

Public Functions

```
inline ModP()
```

```
inline ModP(IntT val)
```

```
inline IntT to_int() const
```

```
inline ModP operator+(const ModP &b) const
```

```
inline ModP &operator+=(const ModP &b)
```

```
inline ModP operator-(const ModP &b) const
```

```
inline ModP &operator-=(const ModP &b)
```

```
inline ModP operator-() const
```

```
inline ModP operator*(const ModP &b) const
```

```
inline ModP &operator*=(const ModP &b)
```

```
inline bool operator==(const ModP &b) const
```

```
inline bool operator!=(const ModP &b) const
```

```
inline bool operator==(const int b) const
```

```
inline bool operator!=(const int b) const
```

```
inline bool operator<(const ModP &b) const
```

```
ModP inv() const
```

```
inline ModP operator/(const ModP &b) const
```

```
inline ModP &operator/=(const ModP &b)
```

```
inline bool iszero() const
```

Friends

```
inline friend std::ostream &operator<<(std::ostream &os, const ModP &x)
```

Template Class **ModP**< *IntT*, 2 >

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_field.hpp`

Inheritance Relationships

Base Type

- `public AbstractField< ModP< IntT, 2 > >` (*Template Class AbstractField*)

Class Documentation

```
template<typename IntT>
```

```
class ModP<IntT, 2> : public AbstractField<ModP<IntT, 2>>
```

Public Functions

```
inline ModP()
```

```
inline ModP(IntT val)
```

```
inline IntT to_int() const
```

```
inline ModP operator+(const ModP &b) const
```



```
inline ModP &operator+=(const ModP &b)

inline ModP operator-(const ModP &b) const

inline ModP &operator--(const ModP &b)

inline ModP &operator-()

inline ModP operator*(const ModP &b) const

inline ModP &operator*=(const ModP &b)

inline ModP operator/(const ModP &b) const

inline ModP &operator/=(const ModP &b)

inline ModP inv() const

inline bool operator==(const ModP &b) const

inline bool operator!=(const ModP &b) const

inline bool operator==(const int b) const

inline bool operator!=(const int b) const

inline ModP operator=(const int &a)

inline bool iszero() const

inline bool operator<(const ModP &b) const
```

Friends

```
inline friend std::ostream &operator<<(std::ostream &os, const ModP &x)
```

Template Class MorsePairing

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_morse_pairing.hpp`

Class Documentation

template<class **CpxT**>

class **MorsePairing**

Public Functions

inline **MorsePairing**()

inline **MorsePairing**(*CpxT* &C)

inline **MorsePairing**(size_t maxdim)

inline **MorsePairing**(std::vector<size_t> ncells)

inline void **clear**()

inline size_t **maxdim**() const

inline size_t **size**(const size_t dim) const

inline size_t **ncells**(const size_t dim) const

inline bool **is_paired**(size_t dim, size_t i) const

inline bool **set_pair**(size_t dim, size_t i, size_t j)

inline bool **set_pair_edge**(const size_t i, const size_t j, const size_t ei)

template<typename **TR**>

inline bool **set_pair_edge**(const size_t i, const size_t j, const size_t ei, const std::vector<*TR*> &rank)

inline const std::vector<size_t> &**up_paired**(size_t dim) const

inline const std::vector<size_t> &**down_paired**(size_t dim) const

inline std::vector<size_t> **unpaired**(size_t dim) const

template<class ...**Ts**>

```
inline cell_ind add(Ts (&... args))
```

```
template<class ...Ts>  
inline cell_ind add_pair(Ts (&... args))
```

```
template<class ...Ts>  
inline auto faces_begin(Ts (&... args))
```

```
template<class ...Ts>  
inline auto faces_end(Ts (&... args))
```

```
inline CSCMatrix<int, size_t> boundary_csc(const size_t dim) const
```

Friends

```
friend class Filtration
```

Template Class MultiGraph

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_multigraph_multigraph.hpp`

Nested Relationships

Nested Types

- *Struct MultiGraph::Edge*
- *Struct MultiGraph::Node*

Class Documentation

```
template<typename NT, typename ET>
```

```
class MultiGraph
```

Public Functions

```
inline MultiGraph()
```

```
inline Node &add_node(NT &a)
```

```
inline Edge &add_edge(Node &a, Node &b, ET &data)
```

```
struct Edge
```

Public Functions

```
inline Edge(Node &sin, Node &tin, ET &x)
```

```
inline ET *get_data()
```

```
inline void print()
```

Public Members

```
ET *data
```

```
Node *src
```

```
Node *targ
```

```
struct Node
```

Public Functions

```
inline Node(NT &x)
```

```
inline Node(NT *x)
```

```
inline void add_input(Edge *in)
```

```
inline void add_output(Edge *out)
```

```
inline NT *get_data()
```

```
inline void print()
```

Public Members

```
NT *data
```

```
std::vector<Edge*> input
```

```
std::vector<Edge*> output
```

Template Class Rational

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_field.hpp`

Inheritance Relationships

Base Type

- `public AbstractField< Rational< IntT > >` (*Template Class AbstractField*)

Class Documentation

```
template<typename IntT>
```

```
class Rational : public AbstractField<Rational<IntT>>
```

Public Functions

```
inline Rational()
```

```
inline Rational(IntT n0, IntT d0)
```

```
inline Rational(IntT n)
```

```
inline IntT to_int() const
```

```
inline Rational operator+(const Rational &b) const
```

```
inline Rational &operator+=(const Rational &b)
```

```
inline Rational operator-(const Rational &b) const
```

```
inline Rational &operator-=(const Rational &b)
```

```
inline Rational operator-() const
```

```
inline Rational operator*(const Rational &b) const
```

```
inline Rational &operator*=(const Rational &b)
```

```
inline bool operator==(const Rational &b) const
```

```
inline bool operator!=(const Rational &b) const
```

```
inline bool operator==(const int b) const

inline bool operator!=(const int b) const

inline bool operator<(const Rational &b) const

inline Rational inv() const

inline Rational operator/(const Rational &b) const

inline Rational &operator/=(const Rational &b)

inline bool iszero() const
```

Friends

```
inline friend std::ostream &operator<<(std::ostream &os, const Rational &x)
```

Template Class SetVector

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_set_vector.hpp`

Class Documentation

```
template<typename TV, typename TI = size_t>
```

```
class SetVector
```

Public Functions

```
inline SetVector()

inline SetVector(const std::set<key_type> indval)

inline SetVector(const std::vector<TI> &ind, const std::vector<TV> &val)

template<typename IT1, typename IT2>
inline SetVector(IT1 indit, IT2 valit, size_t n)

inline SetVector(const TI i)

inline auto nzbegin() const
```

```
inline auto nzend() const

inline auto nzbegin()

inline auto nzend()

inline size_t nnz() const

inline auto lower_bound(const TI &i)

inline auto lower_bound(const TI &i) const

inline auto upper_bound(const TI &i)

inline auto upper_bound(const TI &i) const

inline auto set(const key_type &k)

inline auto set(const TI ind, const TV val)

template<typename itT>
inline auto replace(itT &it, const key_type &k)

template<typename itT>
inline auto replace(itT &it, const TI ind, const TV val)

template<typename itT>
inline auto replace(itT &it, const TV val)

inline const key_type &lastnz() const

inline TV get(const TI ind) const

inline void permute(const std::vector<size_t> &perm)

template<class SVT>
inline void axpy(const TV &a, const SVT &x)

template<class SVT>
inline void axpy(const TV &a, const SVT &x, const TI &firstind, const TI &lastind)

inline void print() const
```

```
inline void print_row() const

template<typename T>
inline bool operator==(const T &other) const
```

Template Class SparseVector

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_vector.hpp`

Class Documentation

```
template<typename TV, typename TI = size_t>
class SparseVector
```

Public Types

```
using val_type = TV
using tmp_type = std::vector<key_type>
```

Public Functions

```
inline void sort()

inline void reverse_inds(size_t n)
    map indices i -> n-i
inline SparseVector()

inline SparseVector(const std::vector<key_type> &indval)

template<typename TV2>
inline SparseVector<TV2, TI> cast_values(TV2) const

inline SparseVector(const std::vector<TI> &ind, const std::vector<TV> &val)

template<typename TI2>
inline SparseVector(const SparseVector<int, TI2> &other)

inline SparseVector(const std::vector<std::tuple<size_t, int>> &ival)

template<typename IT1, typename IT2>
inline SparseVector(IT1 indit, IT2 valit, size_t n)
```



```
inline SparseVector(const TI i, const TV v = TV(1))
```

```
inline SparseVector(std::initializer_list<TI> ind, std::initializer_list<TV> val)  
    constructor for initializer lists
```

Parameters

- **ind** – nonzero indices
- **val** – nonzero values

```
inline SparseVector(std::string &line)
```

```
inline SparseVector shift_inds(TI shift) const
```

```
template<typename T>  
inline bool operator==(const T &other) const
```

```
template<typename T>  
inline bool operator!=(const T &other) const
```

```
template<typename T>  
inline SparseVector operator[](const T &indset) const
```

```
inline auto nzbegin() const
```

```
inline auto nzend() const
```

```
inline auto nzbegin()
```

```
inline auto nzend()
```

```
inline void clear()
```

```
inline void clear_dealloc()
```

```
inline void clear_zeros()
```

```
inline void clear_inds(const std::vector<bool> &c)  
    clear indices marked true in c i.e. remove entry i if c[i] is true
```

```
inline void clear_inds(const std::vector<bool> &c, std::vector<key_type> &tmp)
```

```
inline size_t nnz() const
```

```
inline std::vector<size_t> nzinds() const
```

```
inline std::vector<TV> nzvals() const

inline std::tuple<std::vector<size_t>, std::vector<TV>> nzs() const

inline auto lower_bound(const TI &i)

inline auto lower_bound(const TI &i) const

inline auto upper_bound(const TI &i)

inline auto upper_bound(const TI &i) const

template<typename itT>
inline auto replace(itT &it, const key_type &k)

template<typename itT>
inline auto replace(itT &it, const TI ind, const TV val)

template<typename itT>
inline auto replace(itT &it, const TV val)

inline TV getval(const size_t i) const

inline TV operator[](size_t i) const

inline auto emplace_back(const key_type &k)

inline auto emplace_back(TI ind, TV val)

inline auto find_last_nz(TI i)

inline const key_type &lastnz() const

inline const key_type &firstnz() const

inline void permute(const std::vector<size_t> &perm)

inline void ipermute(const std::vector<size_t> &perm)

inline SparseVector subvector(const std::vector<size_t> &pind) const

inline SparseVector block(const size_t i0, const size_t i1) const
```

```

inline void set_block(const size_t i0, const size_t i1, const SparseVector &b)

inline void J(const size_t m)

inline void swap_rows(const size_t i, const size_t j)

inline void scale_inplace(const TI i, const TV &c)

inline void erase_last_row_of_matrix(const TI i)

inline void erase_rows_after(const TI m)

inline void erase_all()

inline void erase_initial_rows(const TI ndelete)
    erase the first ndelete rows of the matrix
    decrements all subsequent indices by ndelete

inline void erase_last_rows_of_matrix(const TI i, size_t n_deletions)

inline void erase_for_matrix(const TI i)

inline void mix_rows(const size_t i, const size_t j, const TV &a, const TV &b, const TV &c, const TV &d)

inline void insert_rows(const std::vector<size_t> &r_inds, const std::vector<TV> &r_vals)
    (for matrix nonzero row addition usage) Insert row indices at specified locations(assumes indices of rows is
    in ascending order.) Input: r_inds: the indices of new non-zero elements r_vals: the values of new non-zero
    elements The key idea is to a) first modify all existing non-zero indices b) second add new elements c) sort
    all indices

    For example, if indval has inds [3,5] and r_inds = [1,3,4] then for each element in [3,5], say 3, a) find how
    many indices in r_inds that are smaller than 3, there are 1 b) add up: 3+1 = 4 c) loop over r_inds, we find
    3 < 4, add up 1 => 4+1 = 5; keep going find 4 < 5, add up 1 => 5+1 = 6 So, the row with index 3 finally
    locates at 6.

inline void insert_rows(const std::vector<size_t> &r_inds, size_t ninserted = 0)
    (for matrix zero row addition usage) insert zero row indices at specified locations assumes that list of rows
    is in sorted order

    ninserted is lower bound on number inserted before first element

template<class SVT>
inline size_t nnz_intersection(const SVT &x) const
    calculate the number of non-zeros common to two vectors

    Parameters x – sparse vector for comparison

    Returns ct number of non-zeros common to both this vector and x

template<class SVT>
inline void coeff_intersection(const SVT &x, std::map<TV, size_t> &ct) const
    calculate the number of non-zeros common with c * x where c ranges over any possible value

```

Parameters

- **x** – sparse vector for comparison
- **ct** – map from coefficient *c* to number of identical nonzeros with *c***x* *ct* is not cleared in the function.

```
template<class SVT>  
inline void axpy(const TV &a, const SVT &x, std::vector<key_type> &tmp)
```

```
template<class SVT>  
inline void axpy(const TV &a, const SVT &x)
```

```
inline SparseVector operator+(const SparseVector &other) const
```

```
inline SparseVector operator*(const TV a) const  
    scalar multiplication
```

```
inline TV operator*(const SparseVector &x) const  
    dot product
```

```
template<class SVT>  
inline void axpy(const TV &a, const SVT &x, const TI &firstind, const TI &lastind, std::vector<key_type>  
    &tmp)
```

```
template<class SVT>  
inline void axpy(const SVT &x, const std::vector<TV> &coeff, const std::vector<TI> &inds,  
    std::vector<key_type> &tmp)
```

```
inline void scale_inplace(const std::vector<TV> &coeff)
```

```
inline void scale_inplace(const TV c)
```

```
inline SparseVector scale(const TV c) const
```

```
inline void print() const
```

```
inline void print_row() const
```

```
template<typename IO>  
inline void write(IO &io) const
```

```
inline std::string str()
```

```
inline SparseVector kron(const SparseVector &other, size_t m) const
```

```
inline SparseVector tensor(const SparseVector &other, size_t m) const
```

Public Static Functions

```
static inline SparseVector dense(const TV a, const TI m)
    static constructor that fills vector with m copies of a

static inline SparseVector random(size_t n, double p, int maxval, std::default_random_engine &generator)

static inline SparseVector random(size_t n, double p, int maxval)
```

Template Class UnivariatePolynomial

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_polynomial.hpp`

Class Documentation

```
template<typename T>
class UnivariatePolynomial
```

Public Functions

```
inline UnivariatePolynomial(const std::vector<T> &c)

inline UnivariatePolynomial(std::initializer_list<T> l)

inline UnivariatePolynomial(const T &c0)

inline UnivariatePolynomial()

inline bool is_zero() const

template<typename TI>
inline const T operator[](const TI i) const

template<typename TI>
inline T &operator[](const TI i)

inline size_t dim() const

inline size_t degree() const

inline size_t size() const

inline T leading_coeff() const
```

```
inline T operator()(const T &v) const

template<typename TC>
inline TC operator()(const ColumnMatrix<TC> &A, const TC &v) const

inline UnivariatePolynomial operator-() const

inline UnivariatePolynomial operator+(const UnivariatePolynomial &other) const

inline UnivariatePolynomial &operator+=(const UnivariatePolynomial &other)

inline UnivariatePolynomial &operator-=(const UnivariatePolynomial &other)

inline UnivariatePolynomial operator-(const UnivariatePolynomial &other) const

inline UnivariatePolynomial operator*(const UnivariatePolynomial &other) const

inline std::tuple<UnivariatePolynomial, UnivariatePolynomial> divrem(const UnivariatePolynomial &other)
                                                                    const

inline UnivariatePolynomial operator/(const UnivariatePolynomial &other) const

inline UnivariatePolynomial remainder(const UnivariatePolynomial &other) const

template<typename T2>
inline bool operator==(const T2 &other) const

inline bool operator==(const UnivariatePolynomial &other) const

template<typename T2>
inline bool operator!=(const T2 &other) const

inline bool operator!=(const UnivariatePolynomial &other) const

inline UnivariatePolynomial gcd(const UnivariatePolynomial &other) const

inline bool is_monic() const

inline ColumnMatrix<SparseVector<T>> companion_matrix() const

inline void print()
```

Public Static Functions

static inline *UnivariatePolynomial* **identity**()

static inline *UnivariatePolynomial* **zero**()

static inline *UnivariatePolynomial* **monomial**(size_t d, *T* scale = *T*(1))

Friends

inline friend std::ostream &**operator**<<(std::ostream &os, const *UnivariatePolynomial* &p)

Functions

Function `apply_inverse`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_matrix_interface.hpp`

Function Documentation

MAT **apply_inverse**(*MAT*, *MAT*)

Function `apply_inverse_on_left(L<SI>, L<SI>)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation.hpp`

Function Documentation

L<SI> **apply_inverse_on_left**(*L<SI>* a, *L<SI>* b)

Function `apply_inverse_on_left(U<SI>, U<SI>)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation.hpp`

Function Documentation

$U\langle SI \rangle$ **apply_inverse_on_left**($U\langle SI \rangle$ a, $U\langle SI \rangle$ b)

Function **apply_inverse_on_left**($L\langle SI \rangle$, $A\langle SI \rangle$)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation`

Function Documentation

$A\langle SI \rangle$ **apply_inverse_on_left**($L\langle SI \rangle$ a, $A\langle SI \rangle$ b)

Function **apply_inverse_on_left**($U\langle SI \rangle$, $A\langle SI \rangle$)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation`

Function Documentation

$A\langle SI \rangle$ **apply_inverse_on_left**($U\langle SI \rangle$ a, $A\langle SI \rangle$ b)

Function **apply_inverse_on_right**($L\langle SI \rangle$, $L\langle SI \rangle$)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation`

Function Documentation

$L\langle SI \rangle$ **apply_inverse_on_right**($L\langle SI \rangle$ a, $L\langle SI \rangle$ b)

Function **apply_inverse_on_right**($U\langle SI \rangle$, $U\langle SI \rangle$)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation`

Function Documentation

$U\langle SI \rangle$ **apply_inverse_on_right**($U\langle SI \rangle$ a, $U\langle SI \rangle$ b)

Function `apply_inverse_on_right(A<SI>, L<SI>)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation`

Function Documentation

A<SI> **`apply_inverse_on_right`**(*A<SI>* a, *L<SI>* b)

Function `apply_inverse_on_right(A<SI>, U<SI>)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation`

Function Documentation

A<SI> **`apply_inverse_on_right`**(*A<SI>* a, *U<SI>* b)

Template Function `bats::__ChainComplex(const CpxT&, T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_chain_chain_complex.hpp`

Function Documentation

```
template<typename T, typename CpxT>
inline auto bats::__ChainComplex(const CpxT &X, T)
```

Template Function `bats::__ChainComplex(const CpxT&, const CpxT&, T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_chain_chain_complex.hpp`

Function Documentation

```
template<typename T, typename CpxT>
inline auto bats::__ChainComplex(const CpxT &X, const CpxT &A, T)
```

Template Function `bats::__ChainFunctor`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_multigraph_functors.hpp`

Function Documentation

```
template<typename DT, typename T>  
inline auto bats::__ChainFunctor(const DT &D, T)
```

Template Function `bats::__CochainComplex`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_chain_cochain_complex.hpp`

Function Documentation

```
template<typename T, typename CpxT>  
inline auto bats::__CochainComplex(const CpxT &X, T)
```

Template Function `bats::__FilteredChainComplex`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_chain_filtered_chain_complex.hpp`

Function Documentation

```
template<typename FT, typename T, typename CpxT>  
inline auto bats::__FilteredChainComplex(const Filtration<FT, CpxT> &F, T)
```

Template Function `bats::__ReducedChainComplex`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_basis.hpp`

Function Documentation

```
template<typename T, typename CpxT, typename ...Args>  
inline auto bats::__ReducedChainComplex(const CpxT &F, T, Args... args)
```

Template Function bats::__ReducedCochainComplex

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_cohom_basis.hpp`

Function Documentation

```
template<typename T, typename CpxT, typename ...Args>
inline auto bats::__ReducedCochainComplex(const CpxT &F, T, Args... args)
```

Template Function bats::__ReducedFilteredChainComplex

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_persistence_filtered_basis.hpp`

Function Documentation

```
template<typename FT, typename T, typename CpxT, typename ...Args>
inline auto bats::__ReducedFilteredChainComplex(const Filtration<FT, CpxT> &F, T, Args... args)
```

Template Function bats::add_dimension_recursive_flag(Filtration<T, CpxT>&, const NT&, const size_t, const size_t, const std::vector<size_t>&, std::vector<size_t>&, const T&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_flag.hpp`

Function Documentation

```
template<typename CpxT, typename T, typename NT>
void bats::add_dimension_recursive_flag(Filtration<T, CpxT> &F, const NT &nbrs, const size_t d, const
                                         size_t maxd, const std::vector<size_t> &iter_idxes,
                                         std::vector<size_t> &spx_idxes, const T &t)
```

Template Function bats::add_dimension_recursive_flag(Filtration<T, CpxT>&, const NT&, const size_t, const size_t, const std::vector<size_t>&, std::vector<size_t>&, const T&, bool, const cell_ind&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_flag.hpp`

Function Documentation

```
template<typename CpxT, typename T, typename NT>
void bats::add_dimension_recursive_flag(Filtration<T, CpxT> &F, const NT &nbrs, const size_t d, const
                                         size_t maxd, const std::vector<size_t> &iter_idx,
                                         std::vector<size_t> &spx_idx, const T &t, bool face_paired,
                                         const cell_ind &fi)
```

Template Function bats::add_dimension_recursive_flag(CpxT&, const NT&, const size_t, const size_t, const std::vector<size_t>&, std::vector<size_t>&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_flag.hpp`

Function Documentation

```
template<typename CpxT, typename NT>
void bats::add_dimension_recursive_flag(CpxT &X, const NT &nbrs, const size_t d, const size_t maxd,
                                         const std::vector<size_t> &iter_idx, std::vector<size_t>
                                         &spx_idx)
```

Template Function bats::add_dimension_recursive_flag_extension

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_flag.hpp`

Function Documentation

```
template<typename CpxT, typename T, typename NT>
void bats::add_dimension_recursive_flag_extension(Filtration<T, CpxT> &F, const NT &nbrs, const
                                                  size_t d, const size_t maxd, const std::vector<size_t>
                                                  &iter_idx, std::vector<size_t> &spx_idx, const T
                                                  &t, const size_t index_of_edge,
                                                  std::vector<std::vector<size_t>> &inds)
```

Template Function bats::add_dimension_recursive_flag_unsafe

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_flag.hpp`

Function Documentation

```
template<typename CpxT, typename T, typename NT>
void bats::add_dimension_recursive_flag_unsafe(Filtration<T, CpxT> &F, const NT &nbrs, const size_t
d, const size_t maxd, const std::vector<size_t>
&iter_idxs, std::vector<size_t> &spx_idxs, const T t)
```

Function bats::add_dimension_recursive_nerve

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_nerve.hpp

Function Documentation

```
void bats::add_dimension_recursive_nerve(SimplicialComplex &N, std::vector<size_t> spx, const
bats::Cover &cover, std::vector<size_t> intersection, const size_t
dmax)
```

Template Function bats::add_normal_noise(Matrix<T>&, const T, const T)

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_data_gen.hpp

Function Documentation

```
template<typename T>
Matrix<T> &bats::add_normal_noise(Matrix<T> &X, const T mu = T(0), const T sigma = T(1))
```

Template Function bats::add_normal_noise(Matrix<T>&, unsigned, const T, const T)

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_data_gen.hpp

Function Documentation

```
template<typename T>
Matrix<T> &bats::add_normal_noise(Matrix<T> &X, unsigned seed, const T mu = T(0), const T sigma =
T(1))
```

Template Function `bats::add_normal_noise(DataSet<T>&, const T, const T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_data_gen.hpp`

Function Documentation

```
template<typename T>
inline DataSet<T> &bats::add_normal_noise(DataSet<T> &X, const T mu = T(0), const T sigma = T(1))
```

Template Function `bats::add_normal_noise(DataSet<T>&, unsigned, const T, const T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_data_gen.hpp`

Function Documentation

```
template<typename T>
inline DataSet<T> &bats::add_normal_noise(DataSet<T> &X, unsigned seed, const T mu = T(0), const T
                                     sigma = T(1))
```

Template Function `bats::add_uniform_noise(Matrix<T>&, unsigned, const T, const T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_data_gen.hpp`

Function Documentation

```
template<typename T>
Matrix<T> &bats::add_uniform_noise(Matrix<T> &X, unsigned seed, const T lb = T(0), const T ub = T(1))
```

Template Function `bats::add_uniform_noise(Matrix<T>&, const T, const T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_data_gen.hpp`

Function Documentation

```
template<typename T>
inline Matrix<T> &bats::add_uniform_noise(Matrix<T> &X, const T lb = T(0), const T ub = T(1))
```

Template Function `bats::add_uniform_noise(DataSet<T>&, const T, const T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_data_gen.hpp`

Function Documentation

```
template<typename T>
inline DataSet<T> &bats::add_uniform_noise(DataSet<T> &X, const T lb = T(-1), const T ub = T(1))
```

Function `bats::all_pairs`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_rips.hpp`

Function Documentation

```
std::vector<size_t> bats::all_pairs(const size_t n)
```

Template Function `bats::approx_center`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_landmark.hpp`

Function Documentation

```
template<typename T, typename M>
size_t bats::approx_center(const DataSet<T> &D, const M &dist, size_t k = 0, size_t i0 = 0)
```

Template Function `bats::assign_set_lower`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_cover.hpp`

Function Documentation

```
template<typename T>
int bats::assign_set_lower(const T v, const T min_val, const double bin_width, const size_t nsets)
```

Template Function `bats::assign_set_upper`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_cover.hpp`

Function Documentation

```
template<typename T>
int bats::assign_set_upper(const T v, const T min_val, const double bin_width, const size_t nsets)
```

Template Function `bats::barcode(const Diagram<NT, TM>&, size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TM>
inline auto bats::barcode(const Diagram<NT, TM> &dgm, size_t hdim)
    Compute barcode from diagram of vector spaces and linear maps
    Uses divide and conquer algorithm by default.
```

Template Function `bats::barcode(const Diagram<NT, TM>&, size_t, flags::divide_conquer)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TM>
inline auto bats::barcode(const Diagram<NT, TM> &dgm, size_t hdim, flags::divide_conquer)
    Compute barcode from diagram of vector spaces and linear maps
    Uses divide and conquer algorithm
```

Template Function `bats::barcode(const Diagram<NT, TM>&, size_t, flags::leftward)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TM>
inline auto bats::barcode(const Diagram<NT, TM> &dgm, size_t hdim, flags::leftward)
    Compute barcode from diagram of vector spaces and linear maps
    Uses rightward algorithm (sweeps right to left)
```


Template Function bats::barcode(const Diagram<NT, TM>&, size_t, flags::rightward)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TM>
inline auto bats::barcode(const Diagram<NT, TM> &dgm, size_t hdim, flags::rightward)
    Compute barcode from diagram of vector spaces and linear maps
    Uses leftward algorithm (sweeps left to right)
```

Template Function bats::barcode(const Diagram<NT, std::vector<TM>>&, Args ...)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TM, typename ...Args>
inline auto bats::barcode(const Diagram<NT, std::vector<TM>> &dgm, Args... args)
    Compute barcode in all dimensions of diagram
```

Template Function bats::barcode_equality

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_persistence_barcode.hpp`

Function Documentation

```
template<typename T>
bool bats::barcode_equality(const std::vector<PersistencePair<T>> &ps1, const
    std::vector<PersistencePair<T>> &ps2)
```

Template Function bats::barcode_form_divide_conquer

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TM>
auto bats::barcode_form_divide_conquer(const Diagram<NT, TM> &dgm)
```

Template Function `bats::barcode_form_leftright`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TM>
auto bats::barcode_form_leftright(const Diagram<NT, TM> &dgm)
```

Template Function `bats::barcode_form_rightleft`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TM>
auto bats::barcode_form_rightleft(const Diagram<NT, TM> &dgm)
```

Template Function `bats::barcode_from_barcode_form`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename TN, typename TM>
std::vector<bar> bats::barcode_from_barcode_form(const std::vector<TM> &mat, const Diagram<TN, TM>
&dgm)
```

Function `bats::bars_to_pairs`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
std::vector<PersistencePair<size_t>> bats::bars_to_pairs(const std::vector<bar> &bars, size_t hdim)
```

Template Function `bats::binom_to_inds`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_persistence_union_find.hpp`

Function Documentation

```
template<typename T>
inline std::pair<T, T> bats::binom_to_inds(T k, T N)
    compute the squareform indices of binomial index k inputs: k, index between 0 and N-choose-2 N size of set
    returns: (i,j) where i < j
     $k = N\text{-choose-2} - (N-i)\text{-choose-2} + (j-i-1) (N-i)\text{-choose-2} = N\text{-choose-2} - k - (j-i-1) (N-i)\text{-choose-2} \geq N\text{-choose-2} - k$ 
```

Function `bats::bivariate_cover`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_cover.hpp`

Function Documentation

```
auto bats::bivariate_cover(const std::vector<std::set<size_t>> &cover1, const std::vector<std::set<size_t>>
    &cover2)
```

Template Function `bats::Chain(const CpxT&, T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_chain_chain_complex.hpp`

Function Documentation

```
template<typename T, typename CpxT>
inline auto bats::Chain(const CpxT &X, T)
```

Template Function `bats::Chain(const CpxT&, const CpxT&, T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_chain_chain_complex.hpp`

Function Documentation

```
template<typename T, typename CpxT>
inline auto bats::Chain(const CpxT &X, const CpxT &A, T)
```

Template Function bats::Chain(const CellularMap&, T)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_chain_chain_map.hpp`

Function Documentation

```
template<typename T>
inline auto bats::Chain(const CellularMap &F, T)
```

Template Function bats::Chain(const Filtration<FT, CpxT>&, T)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_chain_filtered_chain_compl`

Function Documentation

```
template<typename FT, typename T, typename CpxT>
inline auto bats::Chain(const Filtration<FT, CpxT> &F, T)
```

Template Function bats::Chain(const Diagram<CpxT, CellularMap>&, T)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_multigraph_functors.hpp`

Function Documentation

```
template<typename CpxT, typename T>
inline auto bats::Chain(const Diagram<CpxT, CellularMap> &D, T)
```

Template Function bats::ChainFunctor(const DT&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_multigraph_functors.hpp`

Function Documentation

```
template<typename TM, typename DT>
Diagram<ChainComplex<TM>, ChainMap<TM>> bats::ChainFunctor(const DT &D)
```

Template Function bats::ChainFunctor(const DT&, TF)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_multigraph_functors.hpp`

Function Documentation

```
template<typename TF, typename DT>
inline auto bats::ChainFunctor(const DT &D, TF)
```

Template Function bats::circle

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_data_gen.hpp`

Function Documentation

```
template<typename T>
DataSet<T> bats::circle(const T rad, const size_t n)
```

Template Function bats::CompleteFlagFiltration

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_flag.hpp`

Function Documentation

```
template<typename T>
std::tuple<SimplicialComplex, Filtration<T, SimplicialComplex>> bats::CompleteFlagFiltration(const
                                                                                               std::vector<size_t>
                                                                                               &edges,
                                                                                               const
                                                                                               std::vector<T>
                                                                                               &t, const
                                                                                               size_t n,
                                                                                               const size_t
                                                                                               maxdim,
                                                                                               const T t0)
```

Template Function `bats::coordinate_projection`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_cover.hpp`

Function Documentation

template<typename **T**>
std::vector<**T**> **bats::coordinate_projection**(const *DataSet*<**T**> &**X**, const size_t **i**)

Function `bats::Cube(size_t, size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_grid.hpp`

Function Documentation

CubicalComplex **bats::Cube**(size_t **m**, size_t **n**)
Cubical complex on 2-dimensional grid on **m** x **n** vertices

Function `bats::Cube(size_t, size_t, size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_grid.hpp`

Function Documentation

CubicalComplex **bats::Cube**(size_t **n1**, size_t **n2**, size_t **n3**)
Cubical complex on 3-dimensional grid on **n1** x **n2** x **n3** vertices

Function `bats::Cube(size_t, size_t, size_t, size_t, size_t, size_t, size_t, size_t, size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_grid.hpp`

Function Documentation

CubicalComplex **bats::Cube**(size_t, size_t, size_t, size_t **i0**, size_t **i1**, size_t **j0**, size_t **j1**, size_t **k0**, size_t **k1**)
Subset of Cubical complex of 3-dimensional grid on **n1** x **n2** x **n3** vertices

This function is useful for constructing zigzags through a grid

Parameters

- **n1** – grid size in 1st index
- **n2** – grid size in 2nd index
- **n3** – grid size in 3rd index
- **i0** – start of first index

- **i1** – end of first index
- **j0** – start of second index
- **j1** – end of second index
- **k0** – start of third index
- **k1** – end of third index

Function bats::CubicalMap

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_complex_cubical_map.hpp`

Function Documentation

inline *CellularMap* bats::**CubicalMap**(const *CubicalComplex* &X, const *CubicalComplex* &Y)

Template Function bats::detail::pivot_coeff

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Function Documentation

template<typename **TVec**>
auto bats::detail::**pivot_coeff**(const *TVec* &a, const *TVec* &b)
Calculate

Returns c coefficient for $a += c*b$

Returns d difference in number of nonzeros of a using this coefficient

Template Function bats::DGLinearFunctor

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_multigraph_functors.hpp`

Function Documentation

template<typename **TM**, typename **DT**>
Diagram<*DGVectorSpace*<*TM*>, *DGLinearMap*<*TM*>> bats::**DGLinearFunctor**(const *DT* &D, int degree = -1)
Functor from topological category to category of differential graded vector spaces
Chain functor is degree = -1 (default) Cochain functor is degree = +1. This is contravariant (reverses arrows)

Template Function `bats::DiscreteMorozovZigzag`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_zigzag_zoo.hpp`

Function Documentation

```
template<typename T, typename M>
std::tuple<Diagram<SimplicialComplex, CellularMap>, std::vector<T>> bats::DiscreteMorozovZigzag(const
                                                    DataSet<T>
                                                    &D,
                                                    const
                                                    M
                                                    &dist,
                                                    T rho,
                                                    size_t
                                                    dmax)
```

Template Function `bats::DiscreteMorozovZigzagSets`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_zigzag_zoo.hpp`

Function Documentation

```
template<typename T, typename M>
std::tuple<Diagram<std::set<size_t>, std::vector<size_t>>, std::vector<T>> bats::DiscreteMorozovZigzagSets(const
                                                                 DataSet<T>
                                                                 &D,
                                                                 const
                                                                 M
                                                                 &dist,
                                                                 T
                                                                 rho)
```

Template Function `bats::dowker_edge_param(const Matrix<T>&, const size_t, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_dowker.hpp`

Function Documentation

```
template<typename T>
T bats::dowker_edge_param(const Matrix<T> &pdist, const size_t i, const size_t j)
```


Template Function `bats::dowker_edge_param(const Matrix<T>&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_dowker.hpp`

Function Documentation

```
template<typename T>
Matrix<T> bats::dowker_edge_param(const Matrix<T> &pdist)
```

Template Function `bats::dowker_filtration_edges`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_dowker_cover.hpp`

Function Documentation

```
template<typename T>
std::vector<filtered_edge<T>> bats::dowker_filtration_edges(const Matrix<T> &pdist, const bats::Cover
                                                         &cover, const T rmax)
```

Template Function `bats::DowkerFiltration(const Matrix<T>&, T, size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_dowker.hpp`

Function Documentation

```
template<typename T>
Filtration<T, SimplicialComplex> bats::DowkerFiltration(const Matrix<T> &pdist, T rmax, size_t dmax)
```

Template Function `bats::DowkerFiltration(const DataSet<T>&, const DataSet<T>&, const M&, T, size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_dowker.hpp`

Function Documentation

```
template<typename T, typename M>
Filtration<T, SimplicialComplex> bats::DowkerFiltration(const DataSet<T> &L, const DataSet<T> &X,
                                                         const M &dist, T rmax, size_t dmax)
```

Template Function `bats::DowkerFiltration(const Matrix<T>&, const bats::Cover&, T, size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_dowker_cover.hpp`

Function Documentation

```
template<typename T>
Filtration<T, SimplicialComplex> bats::DowkerFiltration(const Matrix<T> &pdist, const bats::Cover &cover,
T rmax, size_t dmax)
```

Template Function `bats::DowkerFiltration(const DataSet<T>&, const DataSet<T>&, const M&, const bats::Cover&, T, size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_dowker_cover.hpp`

Function Documentation

```
template<typename T, typename M>
Filtration<T, SimplicialComplex> bats::DowkerFiltration(const DataSet<T> &L, const DataSet<T> &X,
const M &dist, const bats::Cover &cover, T rmax,
size_t dmax)
```

Template Function `bats::EilenbergZilber(const CpxT&, const CpxT&, const size_t, T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_eilenberg_zilber.h`

Function Documentation

```
template<typename CpxT, typename T>
auto bats::EilenbergZilber(const CpxT &X, const CpxT &Y, const size_t maxdim, T)
```

Template Function `bats::EilenbergZilber(const CpxT&, const CpxT&, const CpxT&, const CpxT&, const size_t, T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_eilenberg_zilber.h`

Function Documentation

```
template<typename CpxT, typename T>
auto bats::EilenbergZilber(const CpxT &X, const CpxT &A, const CpxT &Y, const CpxT &B, const size_t
                           maxdim, T)
```

Template Function bats::enclosing_radius

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_metric.hpp`

Function Documentation

```
template<typename T>
T bats::enclosing_radius(const Matrix<T> &D)
    compute the enclosing radius from matrix of pairwise distances
    this is the minimum of the largest entry of each row
```

Template Function bats::euclidean

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_rips.hpp`

Function Documentation

```
template<typename T, typename TI>
T bats::euclidean(TI a, TI b, size_t d)
```

Function bats::extension_perm

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
std::vector<size_t> bats::extension_perm(const std::vector<size_t> &perm, const size_t &length)
```

Template Function bats::extract_basis_indices(const **MT**&, const std::vector<size_t>&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Function Documentation

```
template<typename MT>
std::vector<size_t> bats::extract_basis_indices(const MT &Rk, const std::vector<size_t> &p2ck1)
```

Template Function bats::extract_basis_indices(const MT&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Function Documentation

```
template<typename MT>
std::vector<size_t> bats::extract_basis_indices(const MT &Rk)
```

Template Function bats::extract_dimension

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TM>
auto bats::extract_dimension(const Diagram<NT, std::vector<TM>> &D, size_t k)
    extracts dimension k from a diagram with stacked dimensions
```

Template Function bats::extract_pairs

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_persistence_extract.hpp`

Function Documentation

```
template<typename T, typename MT>
std::vector<PersistencePair<T>> bats::extract_pairs(const MT &R, const std::vector<size_t> &p2c, const
                                                    std::vector<T> &valsk, const std::vector<T> &valsk1,
                                                    size_t k = 0)
```

Extract persistence pairs

Parameters

- **R** – reduced matrix in dimension k
- **p2c** – pivot-to-column map for reduced dimension k+1
- **valsk** – filtration values in dimension k
- **valsk1** – filtration values in dimension k+1
- **k** – (optional) dimension

Function `bats::filtration_iperm`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_filtration.hpp`

Function Documentation

`std::vector<std::vector<size_t>> bats::filtration_iperm(const std::vector<std::vector<size_t>> &perms)`

Template Function `bats::filtration_sortperm(const std::vector<T>&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_filtration.hpp`

Function Documentation

`template<typename T>`
`inline std::vector<size_t> bats::filtration_sortperm(const std::vector<T> &v)`

Template Function `bats::filtration_sortperm(const std::vector<std::vector<T>>&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_filtration.hpp`

Function Documentation

`template<typename T>`
`std::vector<std::vector<size_t>> bats::filtration_sortperm(const std::vector<std::vector<T>> &v)`

Function `bats::find_parent`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_persistence_union_find.hpp`

Function Documentation

`size_t bats::find_parent(std::vector<size_t> &parent, size_t i)`
Find parent of node `i` Performs path compression Implementation that avoids recursive function calls
Modified from https://github.com/stat37411/tda/blob/main/include/union_find.hpp

Template Function `bats::find_perm_from_vector`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
template<typename T>
std::vector<size_t> bats::find_perm_from_vector(const std::vector<T> &v)
```

Template Function `bats::flag_filtration_edges`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_dowker.hpp`

Function Documentation

```
template<typename T>
std::vector<filtered_edge<T>> bats::flag_filtration_edges(const Matrix<T> &pdist, const T rmax)
```

Template Function `bats::FlagComplex`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_flag.hpp`

Function Documentation

```
template<typename CpxT>
CpxT bats::FlagComplex(const std::vector<size_t> &edges, const size_t n, const size_t maxdim)
```

Template Function `bats::FlagFiltration`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_flag.hpp`

Function Documentation

```
template<typename CpxT, typename T>
Filtration<T, CpxT> bats::FlagFiltration(std::vector<filtered_edge<T>> &edges, const size_t n, const size_t
maxdim, const T t0)
```

Template Function `bats::FlagFiltration_extension`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_flag.hpp`

Function Documentation

```
template<typename CpxT, typename T>
auto bats::FlagFiltration_extension(std::vector<filtered_edge<T>> &edges, const size_t n, const size_t
                                maxdim, const T t0)
```

Template Function `bats::force_repel_rp`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_spherical.hpp`

Function Documentation

```
template<typename T>
void bats::force_repel_rp(DataSet<T> &v, T step_max)
```

Template Function `bats::Freudenthal(size_t, size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_grid.hpp`

Function Documentation

```
template<typename CpxT>
CpxT bats::Freudenthal(size_t m, size_t n)
    Freudenthal triangulation of 2-dimensional grid on m x n vertices
```

The grid is indexed in row-major order. The index for vertex (i, j) is $j + n * i$

Parameters

- **m** – number of rows
- **n** – number of columns

Template Function `bats::Freudenthal(size_t, size_t, size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_grid.hpp`

Function Documentation

template<typename **CpxT**>

CpxT **bats::Freudenthal**(size_t n1, size_t n2, size_t n3)

Freudenthal triangulation of 3-dimensional grid on n1 x n2 x n3 vertices

The grid is indexed in row-major order. The index for vertex (i, j, k) is $k + n2 * (j + n1 * i)$

Parameters

- **n1** – grid size in 1st index
- **n2** – grid size in 2nd index
- **n3** – grid size in 3rd index

Template Function `bats::Freudenthal(size_t, size_t, size_t, size_t, size_t, size_t, size_t, size_t, size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_grid.hpp`

Function Documentation

template<typename **CpxT**>

CpxT **bats::Freudenthal**(size_t n1, size_t n2, size_t n3, size_t i0, size_t i1, size_t j0, size_t j1, size_t k0, size_t k1)

Subset of Freudenthal triangulation of 3-dimensional grid on n1 x n2 x n3 vertices

The grid is indexed in row-major order. The index for vertex (i, j, k) is $k + n2 * (j + n1 * i)$

This function is useful for constructing zigzags through a grid

Parameters

- **n1** – grid size in 1st index
- **n2** – grid size in 2nd index
- **n3** – grid size in 3rd index
- **i0** – start of first index
- **i1** – end of first index
- **j0** – start of second index
- **j1** – end of second index
- **k0** – start of third index
- **k1** – end of third index

Template Function bats::Freudenthal(const CubicalComplex&, size_t, size_t, size_t)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_grid.hpp`

Function Documentation

template<typename **CpxT**>

CpxT bats::Freudenthal(const *CubicalComplex* &X, size_t n1, size_t n2, size_t n3)

Freudenthal Triangulation of *CubicalComplex*

Parameters

- **X** – cubical complex
- **n1** – 1st dimension grid size
- **n2** – 2nd dimension grid size
- **n3** – 3rd dimension grid size

Template Function bats::future::find_pivot_complete

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_lu.hpp`

Function Documentation

template<typename **MT**>

std::pair<size_t, size_t> bats::future::find_pivot_complete(const *MT* &A, size_t k)

Template Function bats::future::find_pivot_high(const RandomAccessIterator&, ssize_t, ssize_t)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_util.hpp`

Function Documentation

template<typename **RandomAccessIterator**>

ssize_t bats::future::find_pivot_high(const *RandomAccessIterator* &v, ssize_t start, ssize_t end)

Template Function bats::future::find_pivot_high(const RandomAccessIterator&, ssize_t)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_util.hpp`

Function Documentation

```
template<typename RandomAccessIterator>
inline ssize_t bats::future::find_pivot_high(const RandomAccessIterator &v, ssize_t end)
```

Template Function bats::future::gemm

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_dense.hpp`

Function Documentation

```
template<typename M1, typename M2>
auto bats::future::gemm(const M1 &A, const M2 &B)
```

Template Function bats::future::gemv(const MT&, const VT&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_dense.hpp`

Function Documentation

```
template<typename MT, typename VT>
auto bats::future::gemv(const MT &A, const VT &x)
```

Template Function bats::future::gemv(const MT&, const VT1&, VT2&&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_dense.hpp`

Function Documentation

```
template<typename MT, typename VT1, typename VT2>
VT2 &bats::future::gemv(const MT &A, const VT1 &x, VT2 &&y)
```

Template Function bats::future::l_residual(const MT&, l1&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_lu.hpp`

Function Documentation

```
template<typename MT, typename I1>  
bool bats::future::l_residual(const MT &L, I1 &y)
```

Template Function `bats::future::l_residual(const MT&&, I1&&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_lu.hpp`

Function Documentation

```
template<typename MT, typename I1>  
bool bats::future::l_residual(const MT &&L, I1 &&y)
```

Template Function `bats::future::l_solve(const MT&, I1&, I2&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_lu.hpp`

Function Documentation

```
template<typename MT, typename I1, typename I2>  
bool bats::future::l_solve(const MT &L, I1 &y, I2 &x)
```

Template Function `bats::future::l_solve(const MT1&, const MT2&, I1&, I2&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_lu.hpp`

Function Documentation

```
template<typename MT1, typename MT2, typename I1, typename I2>  
bool bats::future::l_solve(const MT1 &Pt, const MT2 &L, I1 &y, I2 &x)
```

Template Function `bats::future::LU`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_lu.hpp`

Function Documentation

template<typename **MT**>
LUFact<*MT*> bats::future::LU(const *MT* &A0)

Function bats::future::operator<<(std::ostream&, const ElementaryPermutation&)

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_permutation.hpp

Function Documentation

std::ostream &bats::future::operator<<(std::ostream &os, const *ElementaryPermutation* &p)

Function bats::future::operator<<(std::ostream&, const CompositePermutation&)

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_permutation.hpp

Function Documentation

std::ostream &bats::future::operator<<(std::ostream &os, const *CompositePermutation* &p)

Template Function bats::future::unit_lower

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_lu.hpp

Function Documentation

template<typename **MT**>
MT bats::future::unit_lower(const *MT* &A)

Template Function bats::future::upper

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_future_lu.hpp

Function Documentation

template<typename **MT**>
MT bats::future::upper(const *MT* &A)

Function bats::gen_cylinder

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_data_gen.hpp

Function Documentation

DataSet<double> bats::gen_cylinder(const size_t n_len, const size_t n_cir, const double rad = 1.0)

Function bats::get_clearing_inds

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp

Function Documentation

std::vector<size_t> bats::get_clearing_inds(const std::vector<size_t> &p2c)

Template Function bats::get_compression_inds

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp

Function Documentation

template<class **TVec**>
std::vector<bool> bats::get_compression_inds(const *ColumnMatrix*<*TVec*> &R)

Template Function bats::get_dM_ZZ_inds

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_zigzag_zoo.hpp

Function Documentation

template<typename **T**>
std::vector<size_t> bats::get_dM_ZZ_inds(const std::vector<**T**> &hds, const **T** rho, const **T** theta)
determine set of indices to use for dM-ZZ construction

Parameters

- **hds** – hausdorff distance of each point to point set. assume in decreasing order.
- **rho** – rips multiplier - should be >10 [OS15 thm 4.6]
- **theta** – depends on dimension of data. In $[0.5, 1/\sqrt{2})$ [OS15 eq. 1]
 - if unsure, set to 0.5 for finer discretization

Template Function bats::get_m

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_dowker.hpp

Function Documentation

template<typename **T**>
std::vector<**T**> bats::get_m(const *Matrix*<**T**> &pdist, size_t nu)

Template Function bats::get_subset

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_cover.hpp

Function Documentation

template<typename **T**>
DataSet<**T**> bats::get_subset(const *DataSet*<**T**> &X, const std::set<size_t> &ind)

Template Function bats::greedy_landmarks

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_landmark.hpp

Function Documentation

template<typename **T**, typename **M**>
DataSet<**T**> bats::greedy_landmarks(const *DataSet*<**T**> &D, const size_t k, const **M** &dist, const size_t i0 = 0)

Template Function `bats::greedy_landmarks_hausdorff(const DataSet<T>&, const M&, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_landmark.hpp`

Function Documentation

```
template<typename T, typename M>
std::tuple<std::vector<size_t>, std::vector<T>> bats::greedy_landmarks_hausdorff(const DataSet<T> &D,
                                                                              const M &dist, const
                                                                              size_t i0 = 0)
```

Template Function `bats::greedy_landmarks_hausdorff(const DataSet<T>&, const M&, const size_t, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_landmark.hpp`

Function Documentation

```
template<typename T, typename M>
std::tuple<std::vector<size_t>, std::vector<T>> bats::greedy_landmarks_hausdorff(const DataSet<T> &D,
                                                                              const M &dist, const
                                                                              size_t i0, const size_t k)

greedy landmarking D - dataset dist - Metric struct i0 - seed point for landmark set k - number of landmarks
template over data type T and metric M return indices of landmarks in order, hausdorff distance to rest of set
```

Template Function `bats::greedy_landmarks_hausdorff(const Matrix<T>&, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_landmark.hpp`

Function Documentation

```
template<typename T>
std::tuple<std::vector<size_t>, std::vector<T>> bats::greedy_landmarks_hausdorff(const Matrix<T> &pdist,
                                                                              const size_t i0 = 0)
```

Template Function `bats::hausdorff_landmarks`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_landmark.hpp`

Function Documentation

template<typename **T**, typename **M**>
DataSet<**T**> bats::hausdorff_landmarks(const *DataSet*<**T**> &**D**, const *T* r, const *M* &dist, const size_t i0 = 0)

Template Function bats::Hom(const Diagram<ChainComplex<TM>, ChainMap<TM>>&, size_t)

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_multigraph_functors.hpp

Function Documentation

template<typename **TM**>
Diagram<*ReducedChainComplex*<**TM**>, **TM**> bats::Hom(const *Diagram*<*ChainComplex*<**TM**>, *ChainMap*<**TM**>> &**D**, size_t k)

Homology functor for dimension k template over matrix type

Template Function bats::Hom(const Diagram<ChainComplex<TM>, ChainMap<TM>>&, bool)

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_multigraph_functors.hpp

Function Documentation

template<typename **TM**>
Diagram<*ReducedChainComplex*<**TM**>, std::vector<**TM**>> bats::Hom(const *Diagram*<*ChainComplex*<**TM**>, *ChainMap*<**TM**>> &**D**, bool topd = false)

Homology functor in all dimensions template over matrix type

when topd is true, a k-dimensional Chain complex will be assumed to be 0 in dimension k+1 and H_k will be computed.

Assumes that all chain complexes have same dimension.

Parameters

- **D** – diagram of ChainComplexes and ChainMaps
- **topd** – (optional, default: false) if true will compute top dimensional homology.

Template Function bats::Hom(const Diagram<DGVectorSpace<TM>, DGLinearMap<TM>>&, size_t)

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_multigraph_functors.hpp

Function Documentation

```
template<typename TM>
Diagram<ReducedDGVectorSpace<TM>, TM> bats::Hom(const Diagram<DGVectorSpace<TM>,
DGLinearMap<TM>>> &D, size_t k)
```

Homology functor for dimension k template over matrix type

Template Function `bats::Hom(const Diagram<DGVectorSpace<TM>, DGLinearMap<TM>>&, bool)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_multigraph_functors.hpp`

Function Documentation

```
template<typename TM>
Diagram<ReducedDGVectorSpace<TM>, std::vector<TM>> bats::Hom(const Diagram<DGVectorSpace<TM>,
DGLinearMap<TM>>> &D, bool topd =
false)
```

Homology functor in all dimensions template over matrix type

when `topd` is true, a k-dimensional Chain complex will be assumed to be 0 in dimension k+1 and H_k will be computed.

Assumes that all chain complexes have same dimension.

Parameters

- **D** – diagram of ChainComplexes and ChainMaps
- **topd** – (optional, default: false) if true will compute top dimensional homology.

Function `bats::identity_perm`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
std::vector<size_t> bats::identity_perm(const size_t &k)
```

Template Function `bats::increment_m`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_dowker.hpp`

Function Documentation

```
template<typename T>
std::vector<T> &bats::increment_m(const Matrix<T> &pdist, std::vector<T> &m)
```

Template Function `bats::induced_map(const ChainMap<ColumnMatrix<TVec>>&, const ReducedChainComplex<ColumnMatrix<TVec>>&, const ReducedChainComplex<ColumnMatrix<TVec>>&, size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_induced_map.hp`

Function Documentation

```
template<class TVec>
ColumnMatrix<TVec> bats::induced_map(const ChainMap<ColumnMatrix<TVec>> &F, const
ReducedChainComplex<ColumnMatrix<TVec>> &C, const
ReducedChainComplex<ColumnMatrix<TVec>> &D, size_t k)
```

Template Function `bats::induced_map(const ChainMap<ColumnMatrix<TVec>>&, const ReducedCochainComplex<ColumnMatrix<TVec>>&, const ReducedCochainComplex<ColumnMatrix<TVec>>&, size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_induced_map.hp`

Function Documentation

```
template<class TVec>
ColumnMatrix<TVec> bats::induced_map(const ChainMap<ColumnMatrix<TVec>> &F, const
ReducedCochainComplex<ColumnMatrix<TVec>> &C, const
ReducedCochainComplex<ColumnMatrix<TVec>> &D, size_t k)
```

obtain induced map on cohomology for dimension k

We assume that F is a chain map, so it is dualized before computing the map

Parameters

- **F** – *ChainMap*
- **C** – ReducedCoChainComplex
- **D** – ReducedCoChainComplex k dimension

Template **Function** **bats::induced_map**(const **DGLinearMap**<**ColumnMatrix**<**TVec**>>&, const **ReducedDGVectorSpace**<**ColumnMatrix**<**TVec**>>&, const **ReducedDGVectorSpace**<**ColumnMatrix**<**TVec**>>&, size_t)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_induced_map.hpp`

Function Documentation

```

template<class TVec>
ColumnMatrix<TVec> bats::induced_map(const DGLinearMap<ColumnMatrix<TVec>> &F, const
                                     ReducedDGVectorSpace<ColumnMatrix<TVec>> &C, const
                                     ReducedDGVectorSpace<ColumnMatrix<TVec>> &D, size_t k)

```

obtain induced map on cohomology for dimension k

We assume that F is a chain map, so it is dualized before computing the map

Parameters

- **F** – *ChainMap*
- **C** – *ReducedCoChainComplex*
- **D** – *ReducedCoChainComplex* k dimension

Template Function bats::interval

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_data_gen.hpp`

Function Documentation

```

template<typename T>
DataSet<T> bats::interval(const T min, const T max, const size_t n)

```

Template Function bats::is_left_arrow

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```

template<typename Edge>
inline bool bats::is_left_arrow(const Edge &e)

```

Template Function `bats::k_choose_2`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_persistence_union_find.hpp`

Function Documentation

```
template<typename T>
inline T bats::k_choose_2(T k)
    compute  $k\text{-choose-2} = k * (k-1) / 2$ 
```

Function `bats::k_choose_2_inv`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_persistence_union_find.hpp`

Function Documentation

```
inline size_t bats::k_choose_2_inv(size_t targ)
    returns next smallest k such that k-choose-2 is  $\geq$  targ
```

Template Function `bats::kdist`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_density.hpp`

Function Documentation

```
template<typename T, typename M>
std::vector<T> bats::kdist(const DataSet<T> &X, const M &dist, const size_t k)
```

Template Function `bats::Kendall_tau(const std::vector<T>&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
template<typename T>
size_t bats::Kendall_tau(const std::vector<T> &perm)
    Kendall-Tau distance between permutation and identity
```

Template Function bats::Kendall_tau(const std::vector<T>&, const std::vector<T>&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
template<typename T>
size_t bats::Kendall_tau(const std::vector<T> &perm1, const std::vector<T> &perm2)
    Kendall-Tau distance between two permutations
```

Template Function bats::Kendall_tau(const std::vector<std::vector<T>>&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
template<typename T>
std::vector<size_t> bats::Kendall_tau(const std::vector<std::vector<T>> &perms)
```

Template Function bats::Kendall_tau_inplace

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

Warning: doxygenfunction: Unable to resolve function “bats::Kendall_tau_inplace” with arguments (TI, int const, int const) in doxygen xml output for project “BATS” from directory: `../doxygen/xml`. Potential matches:

```
- template<typename TI> size_t Kendall_tau_inplace(TI array[], int const begin, int_
  ↳ const end)
```

Template Function bats::Kendall_tau_merge

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

Warning: doxygenfunction: Unable to resolve function “bats::Kendall_tau_merge” with arguments (TI, int const, int const, int const) in doxygen xml output for project “BATS” from directory: ../doxygen/xml. Potential matches:

```
- template<typename TI> size_t Kendall_tau_merge(TI array[], int const left, int,
↳ const mid, int const right)
```

Template Function bats::kron_chain_shift

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_eilenberg_zilber.h

Function Documentation

```
template<typename ChainCpx>
size_t bats::kron_chain_shift(const size_t dx, const ChainCpx &CX, const size_t dy, const ChainCpx &CY)
```

Template Function bats::kron_chains

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_eilenberg_zilber.h

Function Documentation

```
template<typename VT, typename ChainCpx>
auto bats::kron_chains(const VT &cx, const size_t dx, const ChainCpx &CX, const VT &cy, const size_t dy,
const ChainCpx &CY)
```

Template Function bats::kron_homology

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_eilenberg_zilber.h

Function Documentation

```
template<typename ReducedCpx>
auto bats::kron_homology(const size_t dx, const ReducedCpx &RX, const size_t dy, const ReducedCpx &RY,
const ReducedCpx &RXRY)
```

Template Function `bats::kron_index`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_eilenberg_zilber.h`

Function Documentation

```
template<typename CpxT>
std::vector<std::vector<size_t>> bats::kron_index(const CpxT &X, std::vector<std::vector<size_t>> &Ainds,
                                                const CpxT &Y, std::vector<std::vector<size_t>> &Binds,
                                                size_t maxdim)
```

Template Function `bats::landmark_cover`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_cover.hpp`

Function Documentation

```
template<typename T, typename M>
bats::Cover bats::landmark_cover(const DataSet<T> &X, const DataSet<T> &L, const M &dist, size_t k)
```

Template Function `bats::landmark_eps_cover`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_cover.hpp`

Function Documentation

```
template<typename T, typename M>
bats::Cover bats::landmark_eps_cover(const DataSet<T> &X, const DataSet<T> &L, const M &dist, T eps)
```

Function `bats::linear_cover_intersection_diagram`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_inclusion.hpp`

Function Documentation

```
Diagram<std::set<size_t>, std::vector<size_t>> bats::linear_cover_intersection_diagram(std::vector<std::set<size_t>>
                                                                                      &cover)
```

Function `bats::linear_subset_union_diagram`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_inclusion.hpp`

Function Documentation

Diagram`<std::set<size_t>, std::vector<size_t>> bats::linear_subset_union_diagram`(`std::vector<std::set<size_t>>`
&subsets)

Template Function `bats::LowerStarFiltration`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_star.hpp`

Function Documentation

`template<class TC, typename TF>`
Filtration`<TC, TF> bats::LowerStarFiltration`(*TC* &cpx, `std::vector<TF>` f)

Template Function `bats::make_edge`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_rips.hpp`

Function Documentation

`template<typename TF, typename TI>`
inline tedge`<TF, TI> bats::make_edge`(*TI* s, *TI* t, *TF* v)

Template Function `bats::mayer_vietoris_boundary`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_mayer_vietoris.hpp`

Function Documentation

`template<typename CpxT>`
`auto bats::mayer_vietoris_boundary`(const *CpxT* &A, const *CpxT* &X, const *CpxT* &Y, `size_t` k)

Template Function `bats::neighborhood(const VectorView<T>&, const DataSet<T>&, const M&, const T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_neighborhood.hpp`

Function Documentation

```
template<typename T, typename M>
std::vector<size_t> bats::neighborhood(const VectorView<T> &x, const DataSet<T> &X, const M &dist, const
                                     T r)
```

Template Function `bats::neighborhood(const VectorView<T>&, const DataSet<T>&, const M&, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_neighborhood.hpp`

Function Documentation

```
template<typename T, typename M>
std::vector<size_t> bats::neighborhood(const VectorView<T> &x, const DataSet<T> &X, const M &dist, const
                                     size_t k)
```

Template Function `bats::neighborhoods(const DataSet<T>&, const DataSet<T>&, const M&, const T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_neighborhood.hpp`

Function Documentation

```
template<typename T, typename M>
std::vector<std::vector<size_t>> bats::neighborhoods(const DataSet<T> &X, const DataSet<T> &Y, const M
                                                    &dist, const T r)
```

Template Function `bats::neighborhoods(const DataSet<T>&, const DataSet<T>&, const M&, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_neighborhood.hpp`

Function Documentation

```
template<typename T, typename M>
std::vector<std::vector<size_t>> bats::neighborhoods(const DataSet<T> &X, const DataSet<T> &Y, const M
&dist, const size_t k)
```

Template Function `bats::neighborhoods(const Matrix<T>&, const T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_neighborhood.hpp`

Function Documentation

```
template<typename T>
std::vector<std::set<size_t>> bats::neighborhoods(const Matrix<T> &pdist, const T eps = T(0))
```

Function `bats::Nerve(const Diagram<bats::Cover, std::vector<size_t>>&, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_multigraph_functors.hpp`

Function Documentation

```
Diagram<SimplicialComplex, CellularMap> bats::Nerve(const Diagram<bats::Cover, std::vector<size_t>> &D,
const size_t dmax)
```

Function `bats::Nerve(const bats::Cover&, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_nerve.hpp`

Function Documentation

```
SimplicialComplex bats::Nerve(const bats::Cover &cover, const size_t dmax)
```

Template Function `bats::normalize_entries`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_spherical.hpp`

Function Documentation

```
template<typename T>
void bats::normalize_entries(DataSet<T> &data)
```

Template Function bats::operator<

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_rips.hpp`

Function Documentation

```
template<typename TF, typename TI>
inline bool bats::operator<(const tedge<TF, TI> &a, const tedge<TF, TI> &b)
```

Template Function bats::operator<<

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_rips.hpp`

Function Documentation

```
template<typename TF, typename TI>
std::ostream &bats::operator<<(std::ostream &os, tedge<TF, TI> &x)
```

Template Function bats::OscillatingRipsZigzagSets

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_zigzag_zoo.hpp`

Function Documentation

```
template<typename T, typename M>
std::tuple<Diagram<std::set<size_t>, std::vector<size_t>>, std::vector<T>>> bats::OscillatingRipsZigzagSets(const
                                                                    DataSet<T>
                                                                    &D,
                                                                    const
                                                                    M
                                                                    &dist,
                                                                    T
                                                                    rho,
                                                                    T
                                                                    eta)
```

Construct diagram of sets with a vector of Rips parameters

Creates this for Oscillating Rips Zigzag construction

Ref Oudot-Sheehy '15

Parameters

- **D** – data set
- **dist** – metric
- **rho** – multiplier
- **eta** – multiplier. Must have $\eta \leq \rho$

Template Function `bats::pairwise_dist`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_rips.hpp`

Function Documentation

```
template<typename T>
std::vector<T> bats::pairwise_dist(const std::vector<T> &x, const std::vector<size_t> &edges, const size_t d)
```

Template Function `bats::partial_reduce_parallel(ColumnMatrix<TVec>&, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_parallel.hpp`

Function Documentation

```
template<typename TVec>
void bats::partial_reduce_parallel(ColumnMatrix<TVec> &M, const size_t block_size)
    do an initial parallel sweep to zero out columns as possible
```

Parameters

- **M** – matrix to reduce
- **block_size** – size of column blocks to be processed in parallel

Template Function `bats::partial_reduce_parallel(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_parallel.hpp`

Function Documentation

```
template<typename TVec>
void bats::partial_reduce_parallel(ColumnMatrix<TVec> &M, ColumnMatrix<TVec> &U, const size_t
    block_size)
    do an initial parallel sweep to zero out columns as possible
```

Parameters

- **M** – matrix to reduce
- **U** – change of basis

- **block_size** – size of column blocks to be processed in parallel

Template Function `bats::pass_L_left`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TC, typename TM>
void bats::pass_L_left(const Diagram<NT, TM> &dgm, std::vector<SparseFact<TC>> &facts, ssize_t i)
```

Template Function `bats::pass_P_left`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TC, typename TM>
void bats::pass_P_left(const Diagram<NT, TM> &dgm, std::vector<SparseFact<TC>> &facts, ssize_t i)
```

Template Function `bats::pass_P_right`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TC, typename TM>
void bats::pass_P_right(const Diagram<NT, TM> &dgm, std::vector<SparseFact<TC>> &facts, ssize_t i)
```

Template Function `bats::pass_PL_left`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TC, typename TM>
void bats::pass_PL_left(const Diagram<NT, TM> &dgm, std::vector<SparseFact<TC>> &facts,
                        std::vector<TM> &mats, ssize_t i)
```

Template Function `bats::pass_U_right`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TC, typename TM>
void bats::pass_U_right(const Diagram<NT, TM> &dgm, std::vector<SparseFact<TC>> &facts, ssize_t i)
```

Template Function `bats::pass_UP_right`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TC, typename TM>
void bats::pass_UP_right(const Diagram<NT, TM> &dgm, std::vector<SparseFact<TC>> &facts,
                        std::vector<TM> &mats, ssize_t i)
```

Function `bats::perm_to_the_end(const size_t&, const size_t&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
std::vector<size_t> bats::perm_to_the_end(const size_t &index, const size_t &length)
```

Function `bats::perm_to_the_end(const std::vector<size_t>&, const size_t&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
std::vector<size_t> bats::perm_to_the_end(const std::vector<size_t> &index_list, const size_t &length)
```

Template Function `bats::print_1D_vectors`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_print.hpp`

Function Documentation

```
template<typename T>  
void bats::print_1D_vectors(const T &perm)
```

Template Function `bats::print_2D_vectors`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_print.hpp`

Function Documentation

```
template<typename T>  
void bats::print_2D_vectors(const T &perms)
```

Template Function `bats::print_filtration_info`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_print.hpp`

Function Documentation

```
template<class Filtration>  
void bats::print_filtration_info(const Filtration &F)
```

Template Function `bats::print_simplex`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_print.hpp`

Function Documentation

```
template<typename T>  
void bats::print_simplex(const T &perm)
```

Template Function `bats::print_summary_of_filtration`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_print.hpp`

Function Documentation

```
template<typename CpxT, typename T>
void bats::print_summary_of_filtration(const CpxT &X, std::function<T(const std::vector<size_t>&)>
                                     &filtfn)
```

Function `bats::prod_ind`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_complex_simplicial_complex.hpp`

Function Documentation

```
inline size_t bats::prod_ind(size_t i, size_t j, size_t n)
```

Template Function `bats::product_paths(CpxT&, itT, const itT, itT, const itT, std::vector<size_t>&, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_complex_simplicial_complex.hpp`

Function Documentation

```
template<typename CpxT, typename itT>
void bats::product_paths(CpxT &XY, itT xit, const itT xend, itT yit, const itT yend, std::vector<size_t> &s,
                        const size_t n)
```

Template Function `bats::product_paths(CpxT&, itT, const itT, itT, const itT, std::vector<size_t>&, const size_t, std::vector<cell_ind>&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_complex_simplicial_complex.hpp`

Function Documentation

```
template<typename CpxT, typename itT>
void bats::product_paths(CpxT &XY, itT xit, const itT xend, itT yit, const itT yend, std::vector<size_t> &s,
                        const size_t n, std::vector<cell_ind> &ci)
```


Template Function `bats::product_space`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_data_gen.hpp`

Function Documentation

template<typename **T**>
DataSet<*T*> **bats::product_space**(const *DataSet*<*T*> &X, const *DataSet*<*T*> &Y)

Template Function `bats::random_landmarks`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_landmark.hpp`

Function Documentation

template<typename **T**>
DataSet<*T*> **bats::random_landmarks**(const *DataSet*<*T*> &D, const size_t k)

Template Function `bats::read_point_cloud`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_data.hpp`

Function Documentation

template<typename **T** = double>
Matrix<*T*> **bats::read_point_cloud**(std::string &fname, bool header = false)

Template Function `bats::Reduce(const CpxT&, T, Args ...)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_basis.hpp`

Function Documentation

template<typename **T**, typename **CpxT**, typename ...**Args**>
inline auto **bats::Reduce**(const *CpxT* &F, *T*, *Args*... args)

Template Function `bats::Reduce(const ChainComplex<MT>&, Args ...)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_basis.hpp`

Function Documentation

```
template<typename MT, typename ...Args>
inline auto bats::Reduce(const ChainComplex<MT> &C, Args... args)
```

Template Function `bats::Reduce(const Filtration<FT, CpxT>&, T, Args ...)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_persistence_filtered_basis.h`

Function Documentation

```
template<typename FT, typename T, typename CpxT, typename ...Args>
inline auto bats::Reduce(const Filtration<FT, CpxT> &F, T, Args... args)
```

Template Function `bats::Reduce(const FilteredChainComplex<T, MT>&, Args ...)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_persistence_filtered_basis.h`

Function Documentation

```
template<typename T, typename MT, typename ...Args>
inline auto bats::Reduce(const FilteredChainComplex<T, MT> &C, Args... args)
```

Template Function `bats::reduce_block_dq(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&, const size_t, const size_t, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_parallel.hpp`

Function Documentation

```
template<typename TVec>
std::tuple<std::unordered_map<size_t, size_t>, std::vector<size_t>> bats::reduce_block_dq(ColumnMatrix<TVec>
&M, ColumnMatrix<TVec> &U,
const size_t j0,
const size_t j1,
const size_t
max_block_size)
```

reduce a block of columns via divide and conquer

Parameters

- **M** – matrix to be reduced
- **U** – change of basis matrix
- **j0** – start of column range
- **j1** – upper bound of column range
- **max_block_size** – maximum block size for sequential base case

Returns p2c map from pivots to columns for this block

Returns nzcol vector of nonzero columns after reduction

Template Function `bats::reduce_block_dq(ColumnMatrix<TVec>&, const size_t, const size_t, const size_t, const size_t, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_parallel.hpp`

Function Documentation

```
template<typename TVec>
std::tuple<std::unordered_map<size_t, size_t>, std::vector<size_t>> bats::reduce_block_dq(ColumnMatrix<TVec>
&M, const size_t
j0, const size_t j1,
const size_t
max_block_size,
const size_t level,
const size_t
max_depth)
```

reduce a block of columns via divide and conquer

Parameters

- **M** – matrix to be reduced
- **j0** – start of column range
- **j1** – upper bound of column range
- **max_block_size** – maximum block size for sequential base case

Returns p2c map from pivots to columns for this block

Returns nzcol vector of nonzero columns after reduction

Template Function `bats::reduce_block_sequential(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&, const size_t, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_parallel.hpp`

Function Documentation

```
template<typename TVec>
std::tuple<std::unordered_map<size_t, size_t>, std::vector<size_t>> bats::reduce_block_sequential(ColumnMatrix<TVec>
&M,
Column-
Ma-
trix<TVec>
&U,
const
size_t j0,
const
size_t
j1)
```

reduce a block of columns sequentially

Parameters

- **M** – matrix to be reduced
- **U** – change of basis matrix
- **j0** – start of column range
- **j1** – upper bound of column range

Returns p2c map from pivots to columns for this block

Returns nzcol vector of nonzero columns after reduction

Template Function `bats::reduce_block_sequential(ColumnMatrix<TVec>&, const size_t, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_parallel.hpp`

Function Documentation

```
template<typename TVec>
std::tuple<std::unordered_map<size_t, size_t>, std::vector<size_t>> bats::reduce_block_sequential(ColumnMatrix<TVec>
&M,
const
size_t j0,
const
size_t
j1)
```

reduce a block of columns sequentially

Parameters

- **M** – matrix to be reduced
- **j0** – start of column range
- **j1** – upper bound of column range

Returns p2c map from pivots to columns for this block

Returns nzcol vector of nonzero columns after reduction

Template Function `bats::reduce_column_extra`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_parallel.hpp`

Function Documentation

```
template<typename TVec>
void bats::reduce_column_extra(ColumnMatrix<TVec> &M, const size_t j, std::unordered_map<size_t,
                                size_t> &p2c, typename TVec::tmp_type &tmp)
    reduce a single column of the matrix M continue eliminating entries after finding pivot
```

assumes that p2c only contains columns < j

Parameters

- **M** – matrix
- **j** – column to reduce
- **p2c** – map from pivots to columns
- **tmp** – preallocated for faster axpys

Template Function `bats::reduce_column_sparsify`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Function Documentation

```
template<typename TVec, typename F>
void bats::reduce_column_sparsify(ColumnMatrix<TVec> &M, const size_t j, std::vector<size_t>
                                &pivot_to_col, std::map<F, size_t> &coeff, typename TVec::tmp_type
                                &tmp)
    greedily introduce sparsity into column j of M using columns k < j has the effect of reducing column j if not
    already reduced.
```

Parameters

- **M** – matrix
- **j** – column to reduce
- **pivot_to_col** – maps pivots to columns
- **coeff** – preallocated map to use when sparsifying.
- **tmp** – preallocated to use with axpy

Template Function `bats::reduce_column_standard(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&, const size_t, std::unordered_map<size_t, size_t>&, typename TVec::tmp_type&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_parallel.hpp`

Function Documentation

```
template<typename TVec>
void bats::reduce_column_standard(ColumnMatrix<TVec> &M, ColumnMatrix<TVec> &U, const size_t j,
                                std::unordered_map<size_t, size_t> &p2c, typename TVec::tmp_type
                                &tmp)
    reduce a single column of the matrix M
```

assumes that `p2c` only contains columns `< j`

Parameters

- **M** – matrix
- **U** – change of basis matrix
- **j** – column to reduce
- **p2c** – map from pivots to columns
- **tmp** – preallocated for faster axpys

Template Function `bats::reduce_column_standard(ColumnMatrix<TVec>&, const size_t, std::unordered_map<size_t, size_t>&, typename TVec::tmp_type&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_parallel.hpp`

Function Documentation

```
template<typename TVec>
void bats::reduce_column_standard(ColumnMatrix<TVec> &M, const size_t j, std::unordered_map<size_t,
                                size_t> &p2c, typename TVec::tmp_type &tmp)
    reduce a single column of the matrix M
```

assumes that `p2c` only contains columns `< j`

Parameters

- **M** – matrix
- **j** – column to reduce
- **p2c** – map from pivots to columns
- **tmp** – preallocated for faster axpys

Template Function bats::reduce_matrix(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&, divide_conquer_flag)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_parallel.hpp`

Function Documentation

```
template<class TVec>
inline std::vector<size_t> bats::reduce_matrix(ColumnMatrix<TVec> &M, ColumnMatrix<TVec> &U,
                                              divide_conquer_flag)
```

Template Function bats::reduce_matrix(ColumnMatrix<TVec>&, divide_conquer_flag)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_parallel.hpp`

Function Documentation

```
template<class TVec>
inline std::vector<size_t> bats::reduce_matrix(ColumnMatrix<TVec> &M, divide_conquer_flag)
```

Template Function bats::reduce_matrix(ColumnMatrix<TVec>&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Function Documentation

```
template<class TVec>
inline std::vector<size_t> bats::reduce_matrix(ColumnMatrix<TVec> &M)
```

Template Function bats::reduce_matrix(ColumnMatrix<TVec>&, bats::standard_reduction_flag)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Function Documentation

```
template<class TVec>
inline std::vector<size_t> bats::reduce_matrix(ColumnMatrix<TVec> &M, bats::standard_reduction_flag)
```

Template Function `bats::reduce_matrix(ColumnMatrix<TVec>&, bats::extra_reduction_flag)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Function Documentation

```
template<class TVec>
inline std::vector<size_t> bats::reduce_matrix(ColumnMatrix<TVec> &M, bats::extra_reduction_flag)
```

Template Function `bats::reduce_matrix(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Function Documentation

```
template<class TVec>
inline std::vector<size_t> bats::reduce_matrix(ColumnMatrix<TVec> &M, ColumnMatrix<TVec> &U)
```

Template Function `bats::reduce_matrix(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&, bats::standard_reduction_flag)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Function Documentation

```
template<class TVec>
inline std::vector<size_t> bats::reduce_matrix(ColumnMatrix<TVec> &M, ColumnMatrix<TVec> &U,
                                             bats::standard_reduction_flag)
```

Template Function `bats::reduce_matrix(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&, bats::extra_reduction_flag)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Function Documentation

```
template<class TVec>
inline std::vector<size_t> bats::reduce_matrix(ColumnMatrix<TVec> &M, ColumnMatrix<TVec> &U,
                                             bats::extra_reduction_flag)
```


Template	Function	<code>bats::reduce_matrix_clearing(ColumnMatrix<TVec>&,</code>	<code>const</code>
		<code>std::vector<size_t>&, flag)</code>	

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Function Documentation

```
template<class TVec, typename flag>
std::vector<size_t> bats::reduce_matrix_clearing(ColumnMatrix<TVec> &M, const std::vector<size_t>
                                              &clear_inds, flag)
```

Template	Function	<code>bats::reduce_matrix_clearing(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&,</code>	
		<code>const ColumnMatrix<TVec>&, const std::vector<size_t>&, flag)</code>	

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Function Documentation

```
template<class TVec, typename flag>
std::vector<size_t> bats::reduce_matrix_clearing(ColumnMatrix<TVec> &M, ColumnMatrix<TVec> &U,
                                              const ColumnMatrix<TVec> &R, const std::vector<size_t>
                                              &p2c, flag)
```

Template	Function	<code>bats::reduce_matrix_compression(ColumnMatrix<TVec>&,</code>	<code>const</code>
		<code>std::vector<bool>&, flag)</code>	

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Function Documentation

```
template<class TVec, typename flag>
std::vector<size_t> bats::reduce_matrix_compression(ColumnMatrix<TVec> &M, const std::vector<bool>
                                              &comp_inds, flag)
```

Template	Function	<code>bats::reduce_matrix_compression(ColumnMatrix<TVec>&, ColumnMa-</code>	
		<code>trix<TVec>&, const std::vector<bool>&, flag)</code>	

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Function Documentation

```
template<class TVec, typename flag>
std::vector<size_t> bats::reduce_matrix_compression(ColumnMatrix<TVec> &M, ColumnMatrix<TVec>
&U, const std::vector<bool> &comp_inds, flag)
```

Template Function bats::reduce_matrix_extra(ColumnMatrix<TVec>&)

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp

Function Documentation

```
template<class TVec>
std::vector<size_t> bats::reduce_matrix_extra(ColumnMatrix<TVec> &M)
```

Template Function bats::reduce_matrix_extra(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&)

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp

Function Documentation

```
template<class TVec>
std::vector<size_t> bats::reduce_matrix_extra(ColumnMatrix<TVec> &M, ColumnMatrix<TVec> &U)
```

Template Function bats::reduce_matrix_standard(ColumnMatrix<TVec>&)

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp

Function Documentation

```
template<class TVec>
std::vector<size_t> bats::reduce_matrix_standard(ColumnMatrix<TVec> &M)
```

Template Function bats::reduce_matrix_standard(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&)

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp

Function Documentation

```
template<class TVec>
std::vector<size_t> bats::reduce_matrix_standard(ColumnMatrix<TVec> &M, ColumnMatrix<TVec> &U)
```

Template Function bats::remove_extra_cycles

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Function Documentation

```
template<class TVec>
void bats::remove_extra_cycles(const ColumnMatrix<TVec> &R, ColumnMatrix<TVec> &U)
    Update change of basis matrix to not be as dense by removing lower grade cycles from  $U$ .

    Let  $j_1 < j_2$ , and  $R[j_1] = 0$  Then, we can add a linear combination of  $U[j_1]$  to  $U[j_2]$  without changing the matrix
    invariant  $B*U = R$ 
```

Parameters

- R – reduced matrix
- U – change of basis matrix

Template Function `bats::Rips(const Diagram<std::set<size_t>, std::vector<size_t>>&, const DataSet<T>&, const M&, const T, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_multigraph_funcutors.hpp`

Function Documentation

```
template<typename T, typename M>
Diagram<SimplicialComplex, CellularMap> bats::Rips(const Diagram<std::set<size_t>, std::vector<size_t>>
    &D, const DataSet<T> &X, const M &dist, const T
    rmax, const size_t dmax)
```

Template Function `bats::Rips(const Diagram<std::set<size_t>, std::vector<size_t>>&, const DataSet<T>&, const M&, const std::vector<T>&, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_multigraph_funcutors.hpp`

Function Documentation

```
template<typename T, typename M>  
Diagram<SimplicialComplex, CellularMap> bats::Rips(const Diagram<std::set<size_t>, std::vector<size_t>>  
                                                    &D, const DataSet<T> &X, const M &dist, const  
                                                    std::vector<T> &rmax, const size_t dmax)
```

Template Function bats::rips_edges(std::vector<T>&, std::vector<size_t>&, std::vector<T>&)

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_rips.hpp

Function Documentation

```
template<typename T>  
void bats::rips_edges(std::vector<T> &x, std::vector<size_t> &edges, std::vector<T> &t)
```

Template Function bats::rips_edges(const DataSet<T>&, const M&, const T)

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_rips.hpp

Function Documentation

```
template<typename T, typename M>  
std::vector<size_t> bats::rips_edges(const DataSet<T> &X, const M &dist, const T rmax)
```

Template Function bats::rips_edges(const Matrix<T>&, const T)

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_rips.hpp

Function Documentation

```
template<typename T>  
std::vector<size_t> bats::rips_edges(const Matrix<T> &pdist, const T rmax)
```

Template Function `bats::rips_filtration_edges(const DataSet<T>&, const M&, const T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_rips.hpp`

Function Documentation

```
template<typename T, typename M>
std::vector<filtered_edge<T>> bats::rips_filtration_edges(const DataSet<T> &X, const M &dist, const T
                                                         rmax)
```

Template Function `bats::rips_filtration_edges(const Matrix<T>&, const T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_rips.hpp`

Function Documentation

```
template<typename T>
std::vector<filtered_edge<T>> bats::rips_filtration_edges(const Matrix<T> &pdist, const T rmax)
```

Template Function `bats::rips_filtration_edges(const DataSet<T>&, const bats::Cover&, const M&, const T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_rips_cover.hpp`

Function Documentation

```
template<typename T, typename M>
std::vector<filtered_edge<T>> bats::rips_filtration_edges(const DataSet<T> &X, const bats::Cover
                                                         &cover, const M &dist, const T rmax)
```

Template Function `bats::rips_union_find_pairs`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_persistence_union_find.hpp`

Function Documentation

```
template<typename T>
std::vector<PersistencePair<T>> bats::rips_union_find_pairs(const std::vector<size_t> &inds, const
                                                         std::vector<T> &vals)
```

Compute the 0-dimensional Rips barcode using the Union-Find algorithm

Inputs: inds: sort indices of n-choose-2 pairwise distances vals: values of n-choose-2 pairwise distances (sorted by inds)

outputs: 0-dimensional persistence pairs

returns vector of PersistencePairs containing 0-dimensional barcode

Template Function `bats::RipsComplex(const DataSet<T>&, const M&, T, size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_rips.hpp`

Function Documentation

```
template<typename CpxT, typename T, typename M>  
CpxT bats::RipsComplex(const DataSet<T> &X, const M &dist, T rmax, size_t dmax)
```

Template Function `bats::RipsComplex(const Matrix<T>&, T, size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_rips.hpp`

Function Documentation

```
template<typename CpxT, typename T>  
CpxT bats::RipsComplex(const Matrix<T> &pdist, T rmax, size_t dmax)
```

Template Function `bats::RipsFiltration(const DataSet<T>&, const M&, T, size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_rips.hpp`

Function Documentation

```
template<typename CpxT, typename T, typename M>  
Filtration<T, CpxT> bats::RipsFiltration(const DataSet<T> &X, const M &dist, T rmax, size_t dmax)
```

Template Function `bats::RipsFiltration(const Matrix<T>&, T, size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_rips.hpp`

Function Documentation

```
template<typename CpxT, typename T>  
Filtration<T, CpxT> bats::RipsFiltration(const Matrix<T> &pdist, T rmax, size_t dmax)
```

Template Function bats::RipsFiltration(const DataSet<T>&, const bats::Cover&, const M&, T, size_t)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_rips_cover.hpp`

Function Documentation

```
template<typename T, typename M>
Filtration<T, SimplicialComplex> bats::RipsFiltration(const DataSet<T> &X, const bats::Cover &cover,
                                                    const M &dist, T rmax, size_t dmax)
```

Template Function bats::RipsFiltration_extension(const DataSet<T>&, const M&, T, size_t)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_rips.hpp`

Function Documentation

```
template<typename CpxT, typename T, typename M>
auto bats::RipsFiltration_extension(const DataSet<T> &X, const M &dist, T rmax, size_t dmax)
```

Template Function bats::RipsFiltration_extension(const Matrix<T>&, T, size_t)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_rips.hpp`

Function Documentation

```
template<typename CpxT, typename T>
auto bats::RipsFiltration_extension(const Matrix<T> &pdist, T rmax, size_t dmax)
```

Template Function bats::RipsHom

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_multigraph_functors.hpp`

Function Documentation

```
template<typename T, typename M, typename FT>
auto bats::RipsHom(const Diagram<std::set<size_t>, std::vector<size_t>> &D, const DataSet<T> &X, const M
                  &dist, const std::vector<T> &rmax, const size_t hdim, FT)
```

Template Function `bats::rowmajor::get_idx(T, T, T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_grid.hpp`

Function Documentation

```
template<typename T>
inline T bats::rowmajor::get_idx(T i, T j, T n)
    Utility for translating to row-major index
```

Template Function `bats::rowmajor::get_idx(T, T, T, T, T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_grid.hpp`

Function Documentation

```
template<typename T>
inline T bats::rowmajor::get_idx(T i, T j, T k, T n2, T n3)
```

Template Function `bats::sample_cube(const size_t, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_data_gen.hpp`

Function Documentation

```
template<typename T>
DataSet<T> bats::sample_cube(const size_t d, const size_t n)
```

Template Function `bats::sample_cube(const size_t, const size_t, unsigned)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_data_gen.hpp`

Function Documentation

```
template<typename T>
DataSet<T> bats::sample_cube(const size_t d, const size_t n, unsigned seed)
```


Template Function `bats::sample_sphere(const size_t, const size_t, unsigned)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_data_gen.hpp`

Function Documentation

template<typename **T**>
DataSet<**T**> **bats::sample_sphere**(const size_t d, const size_t n, unsigned seed)

Template Function `bats::sample_sphere(const size_t, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_data_gen.hpp`

Function Documentation

template<typename **T**>
DataSet<**T**> **bats::sample_sphere**(const size_t d, const size_t n)

Function `bats::serpinski_diagram`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_extras.hpp`

Function Documentation

Diagram<*CellComplex*, *CellularMap*> **bats::serpinski_diagram**(size_t k)

Template Function `bats::SimplicialMap(const CpxT&, const CpxT&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_complex_simplicial_map.hpp`

Function Documentation

template<typename **CpxT**>
inline *CellularMap* **bats::SimplicialMap**(const *CpxT* &X, const *CpxT* &Y)

Template Function `bats::SimplicialMap(const CpxT&, const CpxT&, const std::vector<size_t>&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_complex_simplicial_map.hp`

Function Documentation

template<typename **CpxT**>
CellularMap `bats::SimplicialMap`(const *CpxT* &X, const *CpxT* &Y, const std::vector<size_t> &f0)

Template Function `bats::sort_edges`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_rips.hpp`

Function Documentation

template<typename **T**>
void `bats::sort_edges`(std::vector<size_t> &edges, std::vector<*T*> &v)

Template Function `bats::sort_indexes`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

template<typename **T**>
std::vector<size_t> `bats::sort_indexes`(const std::vector<*T*> &v)

Template Function `bats::sparsify_basis(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&, const size_t, std::map<F, size_t>&, typename TVec::tmp_type&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Function Documentation

template<typename **TVec**, typename **F**>
void `bats::sparsify_basis`(*ColumnMatrix*<*TVec*> &R, *ColumnMatrix*<*TVec*> &U, const size_t j, std::map<*F*, size_t> &coeff, typename *TVec*::tmp_type &tmp)
greedily introduce sparsity into columns j of *U* and R using columns k < j assumes columns k ≤ j are already reduced
objective to greedily minimize is $\text{nnz}(U[j]) + \text{nnz}(R[j])$

Parameters

- **R** – reduced matrix

- **U** – change of basis matrix
- **j** – column
- **coeff** – preallocated map to use when sparsifying.
- **tmp** – preallocated to use with axpy

Template Function `bats::sparsify_basis(ColumnMatrix<TVec>&, ColumnMatrix<TVec>&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Function Documentation

template<typename **TVec**>
void `bats::sparsify_basis`(*ColumnMatrix<TVec>* &R, *ColumnMatrix<TVec>* &U)
greedily introduce sparsity into columns of *U* and R assumes R is already reduced
objective to greedily minimize is $\text{nnz}(U[j]) + \text{nnz}(R[j])$

Parameters

- **R** – reduced matrix
- **U** – change of basis matrix

Template Function `bats::StrictRipsCoverFiltration`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_rips_cover.hpp`

Function Documentation

template<typename **T**, typename **CpxT**, typename **M**>
Filtration<T, CpxT> `bats::StrictRipsCoverFiltration`(const *DataSet<T>* &X, const *bats::Cover* &cover,
const *M* &dist, *T* rmax, size_t dmax)
Strict Cover *Filtration*

Template Function `bats::StrictWitnessComplex`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_dowker.hpp`

Function Documentation

template<typename **T**, typename **M**>
SimplicialComplex `bats::StrictWitnessComplex`(const *DataSet<T>* &X, const *DataSet<T>* &L, const *M*
&dist, const size_t nu, const *T* rmax, const size_t dmax)

Template Function `bats::test_reduce_result`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_persistence_filtered_basis.h`

Function Documentation

```
template<typename T>
bool bats::test_reduce_result(const T &RFCC2, const T &RFCC)
```

Template Function `bats::TriangulatedProduct(const CpxT&, const CpxT&, const size_t, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_complex_simplicial_complex.h`

Function Documentation

```
template<typename CpxT>
CpxT bats::TriangulatedProduct(const CpxT &X, const CpxT &Y, const size_t maxdim, const size_t n)
```

Template Function `bats::TriangulatedProduct(const CpxT&, const CpxT&, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_complex_simplicial_complex.h`

Function Documentation

```
template<typename CpxT>
inline CpxT bats::TriangulatedProduct(const CpxT &X, const CpxT &Y, const size_t maxdim)
```

Template Function `bats::TriangulatedProduct(const CpxT&, const CpxT&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_complex_simplicial_complex.h`

Function Documentation

```
template<typename CpxT>
inline CpxT bats::TriangulatedProduct(const CpxT &X, const CpxT &Y)
```

Template Function `bats::type_A_dq_common`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TC, typename TM>
ssize_t bats::type_A_dq_common(const Diagram<NT, TM> &dgm, std::vector<SparseFact<TC>> &facts,
                               std::vector<TM> &mats, ssize_t j0, ssize_t j1)
```

Template Function `bats::type_A_dq_EL`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TC, typename TM>
void bats::type_A_dq_EL(const Diagram<NT, TM> &dgm, std::vector<SparseFact<TC>> &facts,
                       std::vector<TM> &mats, ssize_t j0, ssize_t j1)
```

Template Function `bats::type_A_dq_EU`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TC, typename TM>
void bats::type_A_dq_EU(const Diagram<NT, TM> &dgm, std::vector<SparseFact<TC>> &facts,
                       std::vector<TM> &mats, ssize_t j0, ssize_t j1)
```

Template Function `bats::type_A_leftright_sweep1`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TC, typename TM>
void bats::type_A_leftright_sweep1(const Diagram<NT, TM> &dgm, std::vector<SparseFact<TC>> &facts,
                                   std::vector<TM> &mats, ssize_t j0, ssize_t j1)
```

Template Function `bats::type_A_leftright_sweep2`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TC, typename TM>
void bats::type_A_leftright_sweep2(const Diagram<NT, TM> &dgm, std::vector<SparseFact<TC>> &facts,
                                   ssize_t j0, ssize_t j1)
```

Template Function `bats::type_A_rightleft_sweep1`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TC, typename TM>
void bats::type_A_rightleft_sweep1(const Diagram<NT, TM> &dgm, std::vector<SparseFact<TC>> &facts,
                                   std::vector<TM> &mats, ssize_t j0, ssize_t j1)
```

Template Function `bats::type_A_rightleft_sweep2`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_quiver_sparse.hpp`

Function Documentation

```
template<typename NT, typename TC, typename TM>
void bats::type_A_rightleft_sweep2(const Diagram<NT, TM> &dgm, std::vector<SparseFact<TC>> &facts,
                                   ssize_t j0, ssize_t j1)
```

Template Function `bats::uniform_interval_cover`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_cover.hpp`

Function Documentation

```
template<typename T>
bats::Cover bats::uniform_interval_cover(const std::vector<T> &x, const size_t nsets)
```

Template Function `bats::union_find_pairs`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_persistence_union_find.hpp`

Function Documentation

```
template<typename T, typename MT>
std::vector<PersistencePair<T>> bats::union_find_pairs(const FilteredChainComplex<T, MT> &F)
    Compute the 0-dimensional barcode using the Union-Find algorithm
    returns vector of PersistencePairs containing 0-dimensional barcode
```

Template Function `bats::util::apply_iperm_swap`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
template<typename T>
inline void bats::util::apply_iperm_swap(std::vector<T> &data, const std::vector<size_t> &perm)
```

Template Function `bats::util::apply_perm(T, std::vector<T2>&, const std::vector<size_t>&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
template<typename T, typename T2>
void bats::util::apply_perm(T begin, std::vector<T2> &tmp, const std::vector<size_t> &perm)
```

Template Function `bats::util::apply_perm(T *, const std::vector<size_t>&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
template<typename T>
void bats::util::apply_perm(T *begin, const std::vector<size_t> &perm)
```

Template Function `bats::util::apply_perm(std::vector<T>&, const std::vector<size_t>&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
template<typename T>
inline void bats::util::apply_perm(std::vector<T> &data, const std::vector<size_t> &perm)
```

Template Function `bats::util::apply_perm_swap`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
template<typename T>
inline void bats::util::apply_perm_swap(std::vector<T> &data, const std::vector<size_t> &perm)
```

Function `bats::util::fill_partial_sortperm`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
void bats::util::fill_partial_sortperm(const std::vector<size_t> &ind, const std::vector<size_t> &perm,
                                       std::vector<size_t> &indperm)
```

Template Function `bats::util::fill_sortperm(const T&, const T&, std::vector<size_t>&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
template<typename T>
void bats::util::fill_sortperm(const T &begin, const T &end, std::vector<size_t> &perm)
```


Template Function bats::util::fill_sortperm(const std::vector<T>&, std::vector<size_t>&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
template<typename T>
inline void bats::util::fill_sortperm(const std::vector<T> &data, std::vector<size_t> &perm)
```

Template Function bats::util::find_sorted_lt

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_sorted.hpp`

Function Documentation

```
template<typename T, typename TI>
size_t bats::util::find_sorted_lt(const TI &begin, const TI &end, const T &v)
```

Template Function bats::util::firstk

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
template<typename TI>
std::vector<size_t> bats::util::firstk(const TI &begin, const TI &end, const size_t k)
```

Template Function bats::util::has_intersect_sorted

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_sorted.hpp`

Function Documentation

```
template<typename C1, typename C2>
bool bats::util::has_intersect_sorted(const C1 &a, const C2 &b)
```

Template Function `bats::util::has_intersect_sorted_lt`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_sorted.hpp`

Function Documentation

```
template<typename T, typename C1, typename C2>
bool bats::util::has_intersect_sorted_lt(const C1 &a, const C2 &b, const T maxval)
```

Template Function `bats::util::intersect_sorted(const C1&, const C2&, std::vector<T>&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_sorted.hpp`

Function Documentation

```
template<typename T, typename C1, typename C2>
void bats::util::intersect_sorted(const C1 &a, const C2 &b, std::vector<T> &c)
```

Template Function `bats::util::intersect_sorted(const C1&, const C2&, std::set<T>&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_sorted.hpp`

Function Documentation

```
template<typename T, typename C1, typename C2>
void bats::util::intersect_sorted(const C1 &a, const C2 &b, std::set<T> &c)
```

Template Function `bats::util::intersect_sorted_lt`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_sorted.hpp`

Function Documentation

```
template<typename T, typename C1, typename C2>
void bats::util::intersect_sorted_lt(const C1 &a, const C2 &b, const T maxval, std::vector<T> &c)
```

Function `bats::util::inv_perm`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
inline std::vector<size_t> bats::util::inv_perm(const std::vector<size_t> &p)
```

Template Function `bats::util::io::parse_argv(const int, char **, const std::string&&, const T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_io.hpp`

Function Documentation

```
template<typename T>
T bats::util::io::parse_argv(const int argc, char **argv, const std::string &&token, const T default_return)
```

Function `bats::util::io::parse_argv(const int, char **, const std::string&&, const std::string)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_io.hpp`

Function Documentation

```
template<>
std::string bats::util::io::parse_argv(const int argc, char **argv, const std::string &&token, const std::string
                                     default_return)
```

Template Function `bats::util::is_degenerate`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_simplex.hpp`

Function Documentation

```
template<typename T>
bool bats::util::is_degenerate(const std::vector<T> &s)
```

Function `bats::util::partial_perm`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

`std::vector<size_t> bats::util::partial_perm(const std::vector<size_t> &ind, const size_t n)`

Template Function `bats::util::perm_inversions`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

`template<typename T>`
`size_t bats::util::perm_inversions(const std::vector<T> &p)`

Template Function `bats::util::perm_sign`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

`template<typename T>`
`inline int bats::util::perm_sign(const std::vector<T> &p)`

Function `bats::util::rand_perm`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

`std::vector<size_t> bats::util::rand_perm(const size_t n)`

Function `bats::util::random_subset`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_set.hpp`

Function Documentation

`std::set<size_t> bats::util::random_subset(const size_t n, const size_t ns)`

Template Function `bats::util::read_simplex`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_simplex.hpp`

Function Documentation

`template<typename T>`
`void bats::util::read_simplex(std::string &line, std::vector<T> &s)`

Template Function `bats::util::set_intersection`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_set.hpp`

Function Documentation

`template<typename C1, typename C2>`
`std::set<size_t> bats::util::set_intersection(const C1 &a, const C2 &b)`

Template Function `bats::util::set_union`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_set.hpp`

Function Documentation

`template<typename CT1, typename CT2>`
`std::set<size_t> bats::util::set_union(const CT1 &s1, const CT2 &s2)`

Template Function `bats::util::simplex_sign`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_simplex.hpp`

Function Documentation

```
template<typename T>  
int bats::util::simplex_sign(std::vector<T> &s)
```

Function `bats::util::sort_ind_by_perm`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
inline void bats::util::sort_ind_by_perm(std::vector<size_t> &ind, const std::vector<size_t> &perm)
```

Function `bats::util::sort_ind_pair_by_perm`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
void bats::util::sort_ind_pair_by_perm(std::vector<size_t> &ind1, std::vector<size_t> &ind2, const  
std::vector<size_t> &perm)
```

Template Function `bats::util::sort_into`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_sorted.hpp`

Function Documentation

```
template<typename T>  
void bats::util::sort_into(const std::vector<T> &x, std::vector<T> &y)
```

Template Function `bats::util::sort_sum_reduce`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_sorted.hpp`

Function Documentation

```
template<typename TI, typename TV>
void bats::util::sort_sum_reduce(std::vector<TI> &ind, std::vector<TV> &val, const size_t offset,
                                std::vector<size_t> &perm, std::vector<TI> &tmpind, std::vector<TV>
                                &tmpval)
```

Template Function `bats::util::sorted_complement`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_sorted.hpp`

Function Documentation

```
template<typename T>
std::vector<T> bats::util::sorted_complement(const std::vector<T> &ind, size_t n)
```

Template Function `bats::util::sortperm(const std::vector<T> &)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
template<typename T>
std::vector<size_t> bats::util::sortperm(const std::vector<T> &data)
```

Template Function `bats::util::sortperm(const TI&, const TI&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
template<typename TI>
std::vector<size_t> bats::util::sortperm(const TI &begin, const TI &end)
    fill a vector that will return a sort permutation on data
    perm[i] is the index of the ith largest element in the container.
```

Template Function `bats::util::sortperm(RAI, RAI, Compare)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
template<typename RAI, class Compare>
std::vector<size_t> bats::util::sortperm(RAI first, RAI last, Compare comp)
    fill a sortperm using custom comparator
```

Parameters

- **first** – random access iterator at beginning of range to sort
- **last** – random access iterator just past the last element of range to sort
- **comp** – comparison function, `comp(a, b)` should return whether a should come before b

Template Function `bats::util::sortperm(const size_t, const size_t, Compare)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
template<class Compare>
std::vector<size_t> bats::util::sortperm(const size_t first, const size_t last, Compare comp)
    fill a sortperm using custom comparator
```

Parameters

- **first** – random access iterator at beginning of range to sort
- **last** – random access iterator just past the last element of range to sort
- **comp** – comparison function, `comp(a, b)` should return whether a should come before b

Template Function `bats::util::stable_sortperm`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
template<typename T>
std::vector<size_t> bats::util::stable_sortperm(const std::vector<T> &data)
```


Template Function `bats::util::to_set`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_set.hpp`

Function Documentation

```
template<typename T>  
std::set<T> bats::util::to_set(const std::vector<T> &v)
```

Template Function `bats::util::top_k`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
template<typename T>  
std::vector<size_t> bats::util::top_k(const std::vector<T> &data, const size_t k)
```

Template Function `bats::util::top_p`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_permutation.hpp`

Function Documentation

```
template<typename T>  
std::vector<size_t> bats::util::top_p(const std::vector<T> &data, const double p)
```

Template Function `bats::util::write_simplex(IO&, std::vector<T>&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_simplex.hpp`

Function Documentation

```
template<typename IO, typename T>  
void bats::util::write_simplex(IO &io, std::vector<T> &s)
```

Template Function `bats::util::write_simplex(IO&, TI&&, TI&&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_simplex.hpp`

Function Documentation

```
template<typename IO, typename TI>
void bats::util::write_simplex(IO &io, TI &&begin, TI &&end)
```

Template Function `bats::vertex_inclusion_map`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_inclusion.hpp`

Function Documentation

```
template<typename T1, typename T2>
std::vector<size_t> bats::vertex_inclusion_map(const T1 &s, const T2 &t)
```

Template Function `bats::witness_edge_param`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_dowker.hpp`

Function Documentation

```
template<typename T, typename M>
Matrix<T> bats::witness_edge_param(const DataSet<T> &X, const DataSet<T> &L, const M &dist, const
                                size_t nu)
```

Template Function `bats::witness_edges(const DataSet<T>&, const DataSet<T>&, const M&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_dowker.hpp`

Function Documentation

```
template<typename T, typename M>
std::vector<size_t> bats::witness_edges(const DataSet<T> &X, const DataSet<T> &L, const M &dist)
```

Template Function `bats::witness_edges(const DataSet<T>&, const DataSet<T>&, const M&, const T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_dowker.hpp`

Function Documentation

```
template<typename T, typename M>
std::vector<size_t> bats::witness_edges(const DataSet<T> &X, const DataSet<T> &L, const M &dist, const T
rmax)
```

Template Function `bats::witness_neighborhoods`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_dowker.hpp`

Function Documentation

```
template<typename T, typename M>
inline auto bats::witness_neighborhoods(const DataSet<T> &X, const DataSet<T> &L, const M &dist, const
size_t nu, const T rmax)
```

Template Function `bats::WitnessComplex(const DataSet<T>&, const DataSet<T>&, const M&, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_dowker.hpp`

Function Documentation

```
template<typename T, typename M>
SimplicialComplex bats::WitnessComplex(const DataSet<T> &X, const DataSet<T> &L, const M &dist, const
size_t dmax)
```

Template Function `bats::WitnessComplex(const DataSet<T>&, const DataSet<T>&, const M&, const size_t, const T, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_dowker.hpp`

Function Documentation

```
template<typename T, typename M>
SimplicialComplex bats::WitnessComplex(const DataSet<T> &X, const DataSet<T> &L, const M &dist, const
size_t nu, const T rmax, const size_t dmax)
```

Template Function bats::WitnessFiltration

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_dowker.hpp`

Function Documentation

```
template<typename T, typename M>
Filtration<T, SimplicialComplex> bats::WitnessFiltration(const DataSet<T> &L, const DataSet<T> &X,
const M &dist, T rmax, size_t dmax)
```

Template Function bats::zigzag::barcode

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_zigzag_filtration.hpp`

Function Documentation

```
template<typename CpxT, typename T, typename FT, typename opt_flag, typename reduction_flag>
auto bats::zigzag::barcode(const ZigzagFiltration<CpxT, T> &F, ssize_t maxdim, FT, opt_flag,
reduction_flag)
```

Template Function bats::zigzag::detail::apply_basis

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_reduction.hpp`

Function Documentation

```
template<typename MT>
void bats::zigzag::detail::apply_basis(MT &A, MT &L, MT &P, const bool prev_dir, const bool dir)
```

Template Function `bats::zigzag::detail::boundary_insertion_map`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_reduction.hpp`

Function Documentation

```
template<typename VecT>
ColumnMatrix<VecT> bats::zigzag::detail::boundary_insertion_map(const std::vector<size_t> &I, const
                                                                size_t i, const VecT &v)
```

Template Function `bats::zigzag::detail::cube_extrema`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_extension.hpp`

Function Documentation

```
template<typename T>
std::pair<T, T> bats::zigzag::detail::cube_extrema(const std::vector<T> &f0, const std::vector<size_t>
                                                  &cube, const size_t n)
    get maximum and minimum value of a function on cube vertices
```

Parameters

- **f0** – function on vertices - in column-major order
- **cube** – representation of cube, can be degenerate
- **n** – length of 3D cube

Template Function `bats::zigzag::detail::cube_val`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_extension.hpp`

Function Documentation

```
template<typename T>
inline T bats::zigzag::detail::cube_val(const std::vector<T> &f0, const size_t i, const size_t j, const size_t
                                         k, const size_t n)
```

Template Function `bats::zigzag::detail::cycle_insertion_map`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_reduction.hpp`

Function Documentation

```
template<typename VectT>
ColumnMatrix<VectT> bats::zigzag::detail::cycle_insertion_map(const std::vector<size_t> &I, const
                                                             size_t j)
```

Function `bats::zigzag::detail::lex_cmp`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_zigzag_filtration.hpp`

Function Documentation

```
bool bats::zigzag::detail::lex_cmp(const bats::SimplicialComplex &X, size_t dimi, size_t i, size_t dimj,
                                   size_t j)
```

Compare simplices in lexicographical order looking at largest vertex first (v_0, \dots, v_p) < (w_0, \dots, w_q) if v_p < w_q

compare simplex i in dimension $dimi$ with simplex j in dimension $dimj$

Parameters

- **X** – simplicial complex
- **dimi** – dimension of first simplex
- **i** – index of first simplex
- **dimj** – dimension of second simplex
- **j** – index of second simplex

Returns true if first simplex < second simplex, false otherwise

Function `bats::zigzag::detail::rlex_cmp`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_zigzag_filtration.hpp`

Function Documentation

```
bool bats::zigzag::detail::rlex_cmp(const bats::SimplicialComplex &X, size_t dimi, size_t i, size_t dimj,
                                    size_t j)
```

Template Function `bats::zigzag::detail::simplex_extrema`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_extension.hpp`

Function Documentation

```
template<typename T, typename Iterator>
std::pair<T, T> bats::zigzag::detail::simplex_extrema(Iterator it, const Iterator &end, const
                                                    std::vector<T> &f0)
```

Template Function `bats::zigzag::detail::updateBars`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_reduction.hpp`

Function Documentation

```
template<typename T, typename MT, typename Map>
void bats::zigzag::detail::updateBars(std::vector<ZigzagPair<T>> &bars, const rfilt_val<T> &fval,
                                     const size_t hdim, MT &E, Map &piv_to_ind, Map &piv_to_ind2)

    incrementally update barcode
    see quiver/sparse.hpp barcode_from_barcode_form
```

Template Function `bats::zigzag::extend_levelset`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_extension.hpp`

Function Documentation

```
template<typename T>
std::vector<std::vector<std::pair<T, T>>> bats::zigzag::extend_levelset(const std::vector<T> &f0, const
                                                                    CubicalComplex &X, const T eps,
                                                                    const size_t n)
```

Extension of zigzag filtration on vertices to a zigzag filtration on a cubical complex. The filtration progresses from low to high values.

Parameters

- **f0** – function on vertices - stored in column-major format
- **X** – *CubicalComplex* - on a vertex set of size n^3
- **eps** – thickened levelset radius
- **n** – length of cube

Returns val vector of vector of pairs holding zigzag filtration values for each cube.

Template Function `bats::zigzag::extend_zigzag_filtration(const std::vector<T>&, const CpxT&, const T)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_extension.hpp`

Function Documentation

template<typename **T**, typename **CpxT**>

auto bats::zigzag::extend_zigzag_filtration(const std::vector<*T*> &f0, const *CpxT* &X, const *T* eps)

Extension of right filtration on 0-cells to right-filtration on all cells in a complex. Right filtration goes from low to high values

A 0-cell *x* enters the filtration at parameter $f_0(x) - \text{eps}$ and is removed from the filtration at parameter $f_0(x) + \text{eps}$

Higher-dimensional cells are present for the interval that all faces are also present.

Parameters

- **f0** – function on 0-cells
- **X** – complex representing topological space
- **eps** – thickened levelset radius

Template Function `bats::zigzag::extend_zigzag_filtration(const std::vector<T>&, const CubicalComplex&, const T, const size_t)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_extension.hpp`

Function Documentation

template<typename **T**>

ZigzagFiltration<*CubicalComplex*, *T*> bats::zigzag::extend_zigzag_filtration(const std::vector<*T*> &f0, const *CubicalComplex* &X, const *T* eps, const size_t n)

Template Function `bats::zigzag::extra_col_reduction`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_reduction.hpp`

Function Documentation

template<class **VecT**>

void bats::zigzag::extra_col_reduction(const size_t j, *ColumnMatrix*<*VecT*> &M, *ColumnMatrix*<*VecT*> &U, const std::vector<size_t> &p2c, typename *VecT*::tmp_type &tmp)

reduce column *j* past the pivot. Heuristic strategy to reduce fill-in during reduction

Template Function `bats::zigzag::prepare_ChainComplex(const ZigzagFiltration<CpxT, T>&, FT)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_zigzag_filtration.hpp`

Function Documentation

```
template<typename CpxT, typename T, typename FT>
auto bats::zigzag::prepare_ChainComplex(const ZigzagFiltration<CpxT, T> &F, FT)
```

Template Filtration<bats::SimplicialComplex, T>&, FT)	Function bats::zigzag::prepare_ChainComplex(const	ZigzagFiltration<bats::SimplicialComplex, T>&, FT)
--	--	---

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_zigzag_filtration.hpp`

Function Documentation

```
template<typename T, typename FT>
auto bats::zigzag::prepare_ChainComplex(const ZigzagFiltration<bats::SimplicialComplex, T> &F, FT)
```

Template Function bats::zigzag::reduce_column(size_t, ColumnMatrix<VecT>&, ColumnMatrix<VecT>&, std::vector<size_t>&, typename VecT::tmp_type&, reduction_flag)
--

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_reduction.hpp`

Function Documentation

```
template<typename VecT, typename reduction_flag>
size_t bats::zigzag::reduce_column(size_t j, ColumnMatrix<VecT> &M, ColumnMatrix<VecT> &U,
                                   std::vector<size_t> &p2c, typename VecT::tmp_type &tmp,
                                   reduction_flag)
```

reduce column `j` past the pivot. Heuristic strategy to reduce fill-in during reduction reduce a column of `M`. Will eliminate pivots in column using columns to the left. If a pivot is shared by column to right, we will continue reduction on the column to the right.

Parameters

- `j` – column to reduce
- `M` – partially reduced matrix
- `U` – basis matrix
- `p2c` – maps pivots to columns for reduction

Returns final column which was updated

Template Function `bats::zigzag::reduce_column(size_t, ColumnMatrix<VecT>&, ColumnMatrix<VecT>&, std::vector<size_t>&, typename VecT::tmp_type&, bats::extra_reduction_flag)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_reduction.hpp`

Function Documentation

```
template<typename VecT>
size_t bats::zigzag::reduce_column(size_t j, ColumnMatrix<VecT> &M, ColumnMatrix<VecT> &U,
                                   std::vector<size_t> &p2c, typename VecT::tmp_type &tmp,
                                   bats::extra_reduction_flag)
```

Template Function `bats::zigzag::zigzag_barcode_reduction`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_zigzag_reduction.hpp`

Function Documentation

```
template<typename MT, typename T, typename opt_flag, typename reduction_flag>
auto bats::zigzag::zigzag_barcode_reduction(const ChainComplex<MT> &C, const
                                             std::vector<rfilt_val<T>> &filt_order, ssize_t maxdim,
                                             opt_flag, reduction_flag)
```

computes zigzag barcode

Computes a zigzag barcode given a chain complex, entry times, and exit times

Computes reduced matrices directly, and assumes that they have already been permuted into the correct order.

TODO: don't compute top dimension homology - assume spurious
TODO: strategy for keeping homology dimensions small when adding a bunch of cells at the same time.

Parameters

- **C** – Chain complex (in zigzag reduction order)
- **filt_order** – order in which zigzag filtration occurs
- **maxdim** – maximum homology dimension to compute. If set to -1, will default to computing all but top dimension.

Template Function `bats::zigzag_levelsets`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_levelset.hpp`

Function Documentation

```
template<typename CpxT, typename T>
auto bats::zigzag_levelsets(const zigzag::ZigzagFiltration<CpxT, T> &X, T eps, T t0, T t1)
    Function which creates a blocked levelset zigzag diagram from a zigzag::ZigzagFiltration

$$f^{-1}\{ -1 \}([s_i, s_{i+1}]) \cap f^{-1}\{ -1 \}([s_i, s_{i+2}]) \cap f^{-1}\{ -1 \}([s_{i+1}, s_{i+2}])$$

    where  $\text{eps} = s_{i+1} - s_i$ 
```

Parameters

- **X** – *zigzag::ZigzagFiltration*
- **eps** – window size
- **t0** – lower bound on first window
- **t1** – upper bound on last window

Template Function bats::zigzag_toplex

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_levelset.hpp`

Function Documentation

```
template<typename T>
zigzag::ZigzagFiltration<CubicalComplex, T> bats::zigzag_toplex(const
    std::vector<std::vector<std::vector<T>>>>
    &img)
```

Create a zigzag filtration from a 3D image.

Extend zigzag filtration from toplices i.e. the maximal cubes correspond to the pixel grid.

Lower-dimensional cube filtration values are extended from cofaces

This means a $n_1 \times n_2 \times n_3$ image will be on a vertex set of size $(n_1 + 1) \times (n_2 + 1) \times (n_3 + 1)$

Template Function block_reduce

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_persistence_filtration.hpp`

Function Documentation

```
template<typename T, class CpxT, typename FT>
std::vector<std::vector<T>>> block_reduce(Filtration<T, CpxT> &F, FT)
```

Template Function `characteristic_matrix`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_polynomial.hpp`

Function Documentation

```
template<typename T>
auto characteristic_matrix(const ColumnMatrix<SparseVector<T>> &A)
```

Function `commute(MAT, MAT)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_matrix_interface.hpp`

Function Documentation

```
MAT commute(MAT, MAT)
```

Function `commute(EL<SI>, L<SI>)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation.hpp`

Function Documentation

```
L<SI> commute(EL<SI> e11, L<SI> l1)
```

Template Function `complete_pairs`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_persistence_filtration.hpp`

Function Documentation

```
template<typename T, class CpxT, typename FT>
void complete_pairs(Filtration<T, CpxT> &F, MorsePairing<CpxT> &M, size_t d, FT)
```

Template Function CU_inplace

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Function Documentation

```
template<class TC>
void CU_inplace(ColumnMatrix<TC> &C, ColumnMatrix<TC> &U)
```

Template Function delete_pivot

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Function Documentation

```
template<class TC>
inline void delete_pivot(const ColumnMatrix<TC> &A, std::map<size_t, std::vector<size_t>> &p2c, size_t j,
                        size_t i0)
```

Template Function EL_L_commute

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Function Documentation

```
template<typename TC>
ColumnMatrix<TC> EL_L_commute(const ColumnMatrix<TC> &E, const ColumnMatrix<TC> &L)
```

Template Function EU_U_commute

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Function Documentation

```
template<typename TC>
inline ColumnMatrix<TC> EU_U_commute(const ColumnMatrix<TC> &EU, const ColumnMatrix<TC> &U)
```

Template Function `extend_filtration`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_extension.hpp`

Function Documentation

```
template<typename CpxT, typename T>  
auto extend_filtration(const CpxT &X, std::function<std::tuple<T, size_t>(const std::vector<size_t>&)> &f)
```

Template Function `extended_euclidean`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_pid.hpp`

Function Documentation

```
template<typename PID>  
std::tuple<PID, PID, PID, PID, PID> extended_euclidean(const PID &a, const PID &b)
```

Template Function `extract_row_scale`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Function Documentation

```
template<typename TC>  
auto extract_row_scale(ColumnMatrix<TC> &E)
```

Template Function `ff_inv`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_field.hpp`

Function Documentation

```
template<typename IntT, unsigned P>  
IntT ff_inv(const IntT val)
```

Template Function `gemv`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_col_matrix.hpp`

Function Documentation

```
template<class TC>  
TC gemv(ColumnMatrix<TC> &A, const TC &x)
```

Template Function `generating_basis`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_rcf.hpp`

Function Documentation

```
template<typename T>  
ColumnMatrix<SparseVector<T>> generating_basis(const ColumnMa-  
trix<SparseVector<UnivariatePolynomial<T>>> &R,  
const ColumnMatrix<SparseVector<T>> &A)
```

Template Function `get_pivots`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Function Documentation

```
template<class TC>  
std::map<size_t, std::vector<size_t>> get_pivots(const ColumnMatrix<TC> &A)
```

Template Function `inv`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Function Documentation

```
template<class TC>  
ColumnMatrix<TC> inv(const ColumnMatrix<TC> &A)
```

Function isprime

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_field.hpp`

Function Documentation

constexpr bool **isprime**(unsigned p)

Template Function L_EL_commute

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Function Documentation

template<typename TC>
inline *ColumnMatrix*<TC> **L_EL_commute**(const *ColumnMatrix*<TC> &L, const *ColumnMatrix*<TC> &EL)

Template Function l_inv

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_col_matrix.hpp`

Function Documentation

template<class TC>
ColumnMatrix<TC> **l_inv**(const *ColumnMatrix*<TC> &L)

Template Function l_solve(const ColumnMatrix<TC>&, const TC&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_col_matrix.hpp`

Function Documentation

template<class TC>
TC **l_solve**(const *ColumnMatrix*<TC> &L, const *TC* &y)

Template Function `l_solve(const ColumnMatrix<TC>&, const ColumnMatrix<TC>&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_col_matrix.hpp`

Function Documentation

```
template<class TC>
ColumnMatrix<TC> l_solve(const ColumnMatrix<TC> &L, const ColumnMatrix<TC> &A)
```

Template Function `LEUP`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Function Documentation

```
template<class TC>
SparseFact<TC> LEUP(const ColumnMatrix<TC> &A)
```

Function `LEUP_Fact`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_matrix_interface.hpp`

Function Documentation

```
std::tuple<MAT, MAT, MAT, MAT> LEUP_Fact(MAT)
```

Function `LEUP_fact`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation.hpp`

Function Documentation

```
std::tuple<L<SI>, EL<SI>, U<SI>, P<SI>> LEUP_fact(A<SI> a)
```

Template Function `LEUP_inplace`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Function Documentation

```
template<class TC>
void LEUP_inplace(SparseFact<TC> &F)
```

Template Function `lower_star_backwards`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_extension.hpp`

Function Documentation

```
template<typename T>
std::vector<T> lower_star_backwards(const std::vector<std::vector<T>> &grad_dgms, const
                                   std::vector<std::vector<int>> &bndinds, const
                                   std::vector<std::vector<size_t>> &imap)
```

Helper function for computing a gradient on function that is extended in lower star filtration from gradient on persistence pairs

`grad_dgms` and `bndinds` should be flattened in each dimension `grad_dgms[dim][k]` is gradient of birth of pair $k/2$
`grad_dgms[dim][k+1]` is gradient of death of pair $k/2$

Parameters

- **grad_dgms** – gradient of persistence diagrams
- **bndinds** – birth-death indices of each pair
- **imap** – map from simplices to critical vertex in filtration

Template Function `lower_star_filtration(const CpxT&, const std::vector<T>&)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_extension.hpp`

Function Documentation

```
template<typename CpxT, typename T>
auto lower_star_filtration(const CpxT &X, const std::vector<T> &f0)
```

Template **Function** **lower_star_filtration(const** **bats::CubicalComplex&**, **const**
std::vector<std::vector<T>>>&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_extension.hpp`

Function Documentation

```
template<typename T>
auto lower_star_filtration(const bats::CubicalComplex &X, const std::vector<std::vector<T>>> &f0)
```

Template **Function** **lower_star_filtration(const** **bats::CubicalComplex&**, **const**
std::vector<std::vector<std::vector<T>>>&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_extension.hpp`

Function Documentation

```
template<typename T>
auto lower_star_filtration(const bats::CubicalComplex &X, const std::vector<std::vector<std::vector<T>>>
&f0)
```

Template Function LQU

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Function Documentation

```
template<class TC>
SparseFact<TC> LQU(const ColumnMatrix<TC> &A)
```

Template Function LQU_inplace

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Function Documentation

```
template<class TC>
void LQU_inplace(SparseFact<TC> &F)
```

Function `matmul(MAT, MAT)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_matrix_interface.hpp`

Function Documentation

MAT **matmul** (*MAT*, *MAT*)

Function `matmul(A<SI>, A<SI>)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation.hpp`

Function Documentation

A<SI> **matmul** (*A<SI>* a, *A<SI>* b)

Function `matmul(D<SI>, D<SI>)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation.hpp`

Function Documentation

D<SI> **matmul** (*D<SI>* a, *D<SI>* b)

Function `matmul(L<SI>, L<SI>)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation.hpp`

Function Documentation

L<SI> **matmul** (*L<SI>* a, *L<SI>* b)

Function `matmul(U<SI>, U<SI>)`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation.hpp`

Function Documentation

$U<SI>$ **matmul** ($U<SI>$ a, $U<SI>$ b)

Function **matmul**($L<SI>$, $U<SI>$)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation`

Function Documentation

$A<SI>$ **matmul** ($L<SI>$ a, $U<SI>$ b)

Template Function **max_cube_val**(const $\text{std::vector}<\text{size_t}>\&$, const $\text{std::vector}<\text{std::vector}<T>>\&$)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_extension.hpp`

Function Documentation

```
template<typename T>
auto max_cube_val(const  $\text{std::vector}<\text{size\_t}>\&$  c, const  $\text{std::vector}<\text{std::vector}<T>>\&$  f0)
    return maximum vertex value on a cube c assume 2 dimensions
```

Template **Function** **max_cube_val**(const $\text{std::vector}<\text{size_t}>\&$, const
 $\text{std::vector}<\text{std::vector}<\text{std::vector}<T>>>\&$)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_filtration_extension.hpp`

Function Documentation

```
template<typename T>
auto max_cube_val(const  $\text{std::vector}<\text{size\_t}>\&$  c, const  $\text{std::vector}<\text{std::vector}<\text{std::vector}<T>>>\&$  f0)
    return maximum vertex value on a cube c assume 3 dimensions
```

Template Function **operator<**(const $TI\&$, const $\text{nzpair}<TI, TV>\&$)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_abstract_vector.hpp`

Function Documentation

```
template<typename TI, typename TV>
bool operator<(const TI &a, const nzpair<TI, TV> &b)
```

Template Function operator<(const nzpair<TI, TV>&, const TI&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_abstract_vector.hpp`

Function Documentation

```
template<typename TI, typename TV>
bool operator<(const nzpair<TI, TV> &a, const TI &b)
```

Template Function operator<(const nzpair<TI, TV>&, const nzpair<TI, TV>&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_abstract_vector.hpp`

Function Documentation

```
template<typename TI, typename TV>
bool operator<(const nzpair<TI, TV> &a, const nzpair<TI, TV> &b)
```

Template Function operator<<(std::ostream&, const nzpair<TI, TV>&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_abstract_vector.hpp`

Function Documentation

```
template<typename TI, typename TV>
std::ostream &operator<<(std::ostream &os, const nzpair<TI, TV> &x)
```

Function operator<<(std::ostream&, L<SI>&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation.hpp`

Function Documentation

`std::ostream &operator<<(std::ostream &out, L<SI> &c)`

Function operator<<(std::ostream&, U<SI>&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation`

Function Documentation

`std::ostream &operator<<(std::ostream &out, U<SI> &c)`

Function operator<<(std::ostream&, EL<SI>&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation`

Function Documentation

`std::ostream &operator<<(std::ostream &out, EL<SI> &c)`

Function operator<<(std::ostream&, EU<SI>&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation`

Function Documentation

`std::ostream &operator<<(std::ostream &out, EU<SI> &c)`

Function operator<<(std::ostream&, ELH<SI>&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation`

Function Documentation

`std::ostream &operator<<(std::ostream &out, ELH<SI> &c)`

Function operator<<(std::ostream&, EUH<SI>&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation`

Function Documentation

std::ostream &**operator**<<(std::ostream &out, *EUH*<*SI*> &c)

Function operator<<(std::ostream&, P<SI>&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation`

Function Documentation

std::ostream &**operator**<<(std::ostream &out, *P*<*SI*> &c)

Function operator<<(std::ostream&, A<SI>&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation`

Function Documentation

std::ostream &**operator**<<(std::ostream &out, *A*<*SI*> &c)

Template Function pivot_ind

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Function Documentation

template<typename *TC*>
inline size_t **pivot_ind**(const *ColumnMatrix*<*TC*> &E, size_t j)

Template Function PLEU

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Function Documentation

```
template<class TC>
SparseFact<TC> PLEU(const ColumnMatrix<TC> &A)
```

Function PLEU_Fact

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_matrix_interface.hpp`

Function Documentation

```
std::tuple<MAT, MAT, MAT, MAT> PLEU_Fact(MAT)
```

Function PLEU_fact

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation.hpp`

Function Documentation

```
std::tuple<P<SI>, L<SI>, EU<SI>, U<SI>> PLEU_fact(A<SI> a)
```

Template Function PLEU_inplace

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Function Documentation

```
template<class TC>
void PLEU_inplace(SparseFact<TC> &F)
```

Template Function PUEL

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Function Documentation

```
template<class TC>
SparseFact<TC> PUEL(const ColumnMatrix<TC> &A)
```

Function PUEL_Fact

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_matrix_interface.hpp`

Function Documentation

```
std::tuple<MAT, MAT, MAT, MAT> PUEL_Fact(MAT)
```

Function PUEL_fact

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation.hpp`

Function Documentation

```
std::tuple<P<SI>, U<SI>, ELH<SI>, L<SI>> PUEL_fact(A<SI> a)
```

Template Function PUEL_inplace

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Function Documentation

```
template<class TC>
void PUEL_inplace(SparseFact<TC> &F)
```

Template Function RCF_basis

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_rcf.hpp`

Function Documentation

```
template<typename T>
ColumnMatrix<SparseVector<T>> RCF_basis(const ColumnMatrix<SparseVector<UnivariatePolynomial<T>>>
&S, const ColumnMatrix<SparseVector<T>> &A, const
ColumnMatrix<SparseVector<T>> &B)
```

Template Function schur(T&, T&, T&, T&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_schur.hpp`

Function Documentation

```
template<typename T>
T schur(T &A, T &B, T &C, T &D)
```

Template Function schur(ColumnMatrix<CT>&, ColumnMatrix<CT>&, ColumnMatrix<CT>&, ColumnMatrix<CT>&)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_schur.hpp`

Function Documentation

```
template<typename CT>
ColumnMatrix<CT> schur(ColumnMatrix<CT> &A, ColumnMatrix<CT> &B, ColumnMatrix<CT> &C,
ColumnMatrix<CT> &D)
```

Template Function schur(const ColumnMatrix<TV>&, const size_t, const size_t, const size_t, const size_t)

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_schur.hpp`

Function Documentation

```
template<typename TV>  
auto schur(const ColumnMatrix<TV> &A, const size_t i0, const size_t i1, const size_t j0, const size_t j1)
```

Template Function **sgn(T)**

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_field.hpp`

Function Documentation

```
template<typename T>  
inline T sgn(T val)
```

Function **sgn(const int&)**

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_pid.hpp`

Function Documentation

```
int sgn(const int &a)
```

Template Function **smith_factorization**

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_pid.hpp`

Function Documentation

```
template<class TC>  
SmithFact<TC> smith_factorization(const ColumnMatrix<TC> &A)
```

Template Function **smith_normal_form**

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_pid.hpp`

Function Documentation

```
template<typename TC>
void smith_normal_form(ColumnMatrix<TC> &A)
```

Template Function smith_rows

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_pid.hpp`

Function Documentation

```
template<class TC>
std::tuple<ColumnMatrix<TC>, ColumnMatrix<TC>> smith_rows(const ColumnMatrix<TC> &A)
```

Template Function standard_reduce

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_persistence_filtration.hpp`

Function Documentation

```
template<typename T, class CpxT, typename FT>
std::vector<std::vector<T>> standard_reduce(const Filtration<T, CpxT> &F, FT)
```

Template Function U_EU_commute

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Function Documentation

```
template<typename TC>
inline ColumnMatrix<TC> U_EU_commute(const ColumnMatrix<TC> &U, const ColumnMatrix<TC> &EU)
```

Template Function u_inv

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_col_matrix.hpp`

Function Documentation

```
template<class TC>
ColumnMatrix<TC> u_inv(const ColumnMatrix<TC> &U)
```

Template Function u_solve(const ColumnMatrix<TC>&, const TC&)

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_col_matrix.hpp

Function Documentation

```
template<class TC>
TC u_solve(const ColumnMatrix<TC> &U, const TC &y)
```

Template Function u_solve(const ColumnMatrix<TC>&, const ColumnMatrix<TC>&)

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_col_matrix.hpp

Function Documentation

```
template<class TC>
ColumnMatrix<TC> u_solve(const ColumnMatrix<TC> &U, const ColumnMatrix<TC> &A)
```

Template Function UELP

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp

Function Documentation

```
template<class TC>
SparseFact<TC> UELP(const ColumnMatrix<TC> &A)
```

Function UELP_Fact

- Defined in file __home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_matrix_interface.hpp

Function Documentation

`std::tuple<MAT, MAT, MAT, MAT> UELP_Fact(MAT)`

Function UELP_fact

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation`

Function Documentation

`std::tuple<U<SI>, EUH<SI>, L<SI>, P<SI>> UELP_fact(A<SI> a)`

Template Function UELP_inplace

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Function Documentation

```
template<class TC>
void UELP_inplace(SparseFact<TC> &F)
```

Template Function update_pivot

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Function Documentation

```
template<class TC>
inline void update_pivot(const ColumnMatrix<TC> &A, std::map<size_t, std::vector<size_t>> &p2c, size_t j,
                        size_t i0)
```

Template Function UQL

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_sparse_fact.hpp`

Function Documentation

```
template<class TC>  
SparseFact<TC> UQL(const ColumnMatrix<TC> &A)
```

Variables

Variable **bats::NO_IND**

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_util_common.hpp`

Variable Documentation

```
const size_t bats::NO_IND = std::numeric_limits<size_t>::max()
```

Defines

Define **APPLY_INVERSE_ON_LEFT**

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation.hpp`

Define Documentation

```
APPLY_INVERSE_ON_LEFT(A, B, C)
```

Define **APPLY_INVERSE_ON_RIGHT**

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation.hpp`

Define Documentation

```
APPLY_INVERSE_ON_RIGHT(A, B, C)
```

Define **IMPL_COUT**

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation.hpp`

Define Documentation**IMPL_COUT**(M)**Define IMPL_FACT**

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation.hpp`

Define Documentation**IMPL_FACT**(TYPE, F1, F2, F3, F4, f1, f2, f3, f4)**Define MM**

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_linalg_symbolic_implementation.hpp`

Define Documentation**MM**(A, B, C)**Define p2c_type**

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_homology_reduction.hpp`

Define Documentation**p2c_type****Typedefs****Typedef bats::Cover**

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_cover.hpp`

Typedef Documentation

```
using bats : Cover = std::vector<std::set<size_t>>
```

Typedef `bats::Matrix`

- Defined in file `__home_docs_checkouts_readthedocs.org_user_builds_bats_checkouts_latest_include_topology_data.hpp`

Typedef Documentation

```
using bats : Matrix = A<Dense<T, RowMaj>>
```

1.6 About BATS

BATS is a library for applied algebraic topology.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

A

A (C++ struct), 28
 AbstractField (C++ class), 75
 AbstractField::inv (C++ function), 75
 AbstractField::operator* (C++ function), 75
 AbstractField::operator+ (C++ function), 75
 AbstractField::operator/ (C++ function), 75
 AbstractField::operator== (C++ function), 75
 AbstractField::operator- (C++ function), 75
 AbstractField::operator<< (C++ function), 76
 AbstractField::print (C++ function), 75
 AbstractField::str (C++ function), 75
 AbstractMatrix (C++ class), 76
 apply_inverse (C++ function), 131
 APPLY_INVERSE_ON_LEFT (C macro), 244
 apply_inverse_on_left (C++ function), 131, 132
 APPLY_INVERSE_ON_RIGHT (C macro), 244
 apply_inverse_on_right (C++ function), 132, 133

B

bats::__ChainComplex (C++ function), 133
 bats::__ChainFunctor (C++ function), 134
 bats::__CochainComplex (C++ function), 134
 bats::__FilteredChainComplex (C++ function), 134
 bats::__ReducedChainComplex (C++ function), 134
 bats::__ReducedCochainComplex (C++ function), 135
 bats::__ReducedFilteredChainComplex (C++ function), 135
 bats::AbstractMetric (C++ struct), 28
 bats::AbstractMetric::dist (C++ function), 28
 bats::AbstractMetric::operator() (C++ function), 28
 bats::add_dimension_recursive_flag (C++ function), 135, 136
 bats::add_dimension_recursive_flag_extension (C++ function), 136
 bats::add_dimension_recursive_flag_unsafe (C++ function), 137
 bats::add_dimension_recursive_nerve (C++ function), 137
 bats::add_normal_noise (C++ function), 137, 138

bats::add_uniform_noise (C++ function), 138, 139
 bats::all_pairs (C++ function), 139
 bats::AngleDist (C++ struct), 29
 bats::AngleDist::dist (C++ function), 29
 bats::apparent_pairs_flag (C++ struct), 29
 bats::approx_center (C++ function), 139
 bats::assign_set_lower (C++ function), 139
 bats::assign_set_upper (C++ function), 140
 bats::bar (C++ struct), 29
 bats::bar::bar (C++ function), 29
 bats::bar::end (C++ member), 30
 bats::bar::end_ind (C++ member), 30
 bats::bar::start (C++ member), 30
 bats::bar::start_ind (C++ member), 30
 bats::barcode (C++ function), 140, 141
 bats::barcode_equality (C++ function), 141
 bats::barcode_form_divide_conquer (C++ function), 141
 bats::barcode_form_leftright (C++ function), 142
 bats::barcode_form_rightleft (C++ function), 142
 bats::barcode_from_barcode_form (C++ function), 142
 bats::BarcodePair (C++ class), 76
 bats::BarcodePair::BarcodePair (C++ function), 76
 bats::BarcodePair::birth (C++ member), 77
 bats::BarcodePair::death (C++ member), 77
 bats::BarcodePair::print (C++ function), 76
 bats::bars_to_pairs (C++ function), 142
 bats::binom_to_inds (C++ function), 143
 bats::bivariate_cover (C++ function), 143
 bats::CellComplex (C++ class), 77
 bats::CellComplex::add (C++ function), 77
 bats::CellComplex::add_vertex (C++ function), 77
 bats::CellComplex::add_vertices (C++ function), 77
 bats::CellComplex::boundary (C++ function), 77
 bats::CellComplex::boundary_csc (C++ function), 77
 bats::CellComplex::CellComplex (C++ function), 77
 bats::CellComplex::maxdim (C++ function), 77

bats::CellComplex::ncells (C++ function), 77
 bats::CellularMap (C++ class), 78
 bats::CellularMap::CellularMap (C++ function), 78
 bats::CellularMap::identity (C++ function), 78
 bats::CellularMap::maxdim (C++ function), 78
 bats::CellularMap::operator[] (C++ function), 78
 bats::CellularMap::save (C++ function), 78
 bats::Chain (C++ function), 143, 144
 bats::ChainComplex (C++ struct), 30
 bats::ChainComplex::boundary (C++ member), 31
 bats::ChainComplex::ChainComplex (C++ function), 30
 bats::ChainComplex::clear_compress_apparent_pairs (C++ function), 31
 bats::ChainComplex::dim (C++ function), 30
 bats::ChainComplex::ipermute_basis (C++ function), 31
 bats::ChainComplex::is_valid_complex (C++ function), 30
 bats::ChainComplex::maxdim (C++ function), 30
 bats::ChainComplex::operator[] (C++ function), 30, 31
 bats::ChainComplex::permute_basis (C++ function), 31
 bats::ChainComplex::relative_complex (C++ function), 30
 bats::ChainComplex::subcomplex (C++ function), 30
 bats::ChainComplex::tensor_product (C++ function), 31
 bats::ChainComplex::update_basis_general (C++ function), 31
 bats::ChainFunctor (C++ function), 145
 bats::ChainMap (C++ struct), 32
 bats::ChainMap::ChainMap (C++ function), 32
 bats::ChainMap::map (C++ member), 33
 bats::ChainMap::maxdim (C++ function), 32
 bats::ChainMap::operator[] (C++ function), 32
 bats::ChainMap::permute_column_basis (C++ function), 32
 bats::ChainMap::permute_row_basis (C++ function), 32
 bats::ChainMap::relative_map (C++ function), 32
 bats::circle (C++ function), 145
 bats::clearing_flag (C++ struct), 33
 bats::CochainComplex (C++ struct), 33
 bats::CochainComplex::coboundary (C++ member), 34
 bats::CochainComplex::CochainComplex (C++ function), 33
 bats::CochainComplex::dim (C++ member), 34
 bats::CochainComplex::ipermute_basis (C++ function), 33
 bats::CochainComplex::maxdim (C++ function), 33
 bats::CochainComplex::operator[] (C++ function), 33
 bats::CochainComplex::permute_basis (C++ function), 33
 bats::CompleteFlagFiltration (C++ function), 145
 bats::compression_flag (C++ struct), 34
 bats::compute_basis_flag (C++ struct), 34
 bats::coordinate_projection (C++ function), 146
 bats::CosineDist (C++ struct), 34
 bats::CosineDist::dist (C++ function), 35
 bats::Cover (C++ type), 246
 bats::Cube (C++ function), 146
 bats::CubicalComplex (C++ class), 79
 bats::CubicalComplex::add (C++ function), 79
 bats::CubicalComplex::add_recursive (C++ function), 79
 bats::CubicalComplex::boundary_csc (C++ function), 80
 bats::CubicalComplex::cell_begin (C++ function), 79
 bats::CubicalComplex::cell_end (C++ function), 80
 bats::CubicalComplex::CubicalComplex (C++ function), 79
 bats::CubicalComplex::faces_begin (C++ function), 79
 bats::CubicalComplex::faces_end (C++ function), 79
 bats::CubicalComplex::find_idx (C++ function), 79
 bats::CubicalComplex::generate_cube (C++ function), 80
 bats::CubicalComplex::get_cell (C++ function), 80
 bats::CubicalComplex::get_cube (C++ function), 80
 bats::CubicalComplex::get_cubes (C++ function), 80
 bats::CubicalComplex::load_cubes (C++ function), 80
 bats::CubicalComplex::maxdim (C++ function), 79
 bats::CubicalComplex::ncells (C++ function), 79
 bats::CubicalComplex::print_summary (C++ function), 79
 bats::CubicalComplex::set_dimension (C++ function), 79
 bats::CubicalComplex::skeleton (C++ function), 80
 bats::CubicalMap (C++ function), 147
 bats::DataSet (C++ struct), 35
 bats::DataSet::data (C++ member), 36
 bats::DataSet::DataSet (C++ function), 35
 bats::DataSet::dim (C++ function), 35

bats::DataSet::get_data (C++ function), 35
bats::DataSet::operator() (C++ function), 35
bats::DataSet::operator[] (C++ function), 35
bats::DataSet::size (C++ function), 35
bats::detail::pivot_coeff (C++ function), 147
bats::DGLinearFunctor (C++ function), 147
bats::DGLinearMap (C++ struct), 36
bats::DGLinearMap::DGLinearMap (C++ function), 36
bats::DGLinearMap::map (C++ member), 37
bats::DGLinearMap::maxdim (C++ function), 36
bats::DGLinearMap::operator[] (C++ function), 36
bats::DGLinearMap::permute_column_basis (C++ function), 36
bats::DGLinearMap::permute_row_basis (C++ function), 36
bats::DGVectorSpace (C++ struct), 37
bats::DGVectorSpace::degree (C++ member), 38
bats::DGVectorSpace::DGVectorSpace (C++ function), 37
bats::DGVectorSpace::differential (C++ member), 38
bats::DGVectorSpace::dim (C++ function), 37
bats::DGVectorSpace::ipermute_basis (C++ function), 38
bats::DGVectorSpace::is_differential (C++ function), 38
bats::DGVectorSpace::maxdim (C++ function), 37
bats::DGVectorSpace::operator[] (C++ function), 37
bats::DGVectorSpace::permute_basis (C++ function), 37, 38
bats::Diagram (C++ class), 81
bats::Diagram::add_edge (C++ function), 81
bats::Diagram::add_node (C++ function), 81
bats::Diagram::Diagram (C++ function), 81
bats::Diagram::edata (C++ member), 82
bats::Diagram::Edge (C++ struct), 38, 82
bats::Diagram::Edge::Edge (C++ function), 38, 82
bats::Diagram::Edge::src (C++ member), 39, 82
bats::Diagram::Edge::targ (C++ member), 39, 82
bats::Diagram::edge_data (C++ function), 81
bats::Diagram::edge_source (C++ function), 81
bats::Diagram::edge_target (C++ function), 81
bats::Diagram::elist (C++ member), 82
bats::Diagram::nedge (C++ function), 81
bats::Diagram::nnode (C++ function), 81
bats::Diagram::node (C++ member), 82
bats::Diagram::node_data (C++ function), 81
bats::Diagram::save (C++ function), 81
bats::Diagram::save_metadata (C++ function), 81
bats::Diagram::set_edge (C++ function), 81
bats::Diagram::set_node (C++ function), 81
bats::DiscreteMorozovZigzag (C++ function), 148
bats::DiscreteMorozovZigzagSets (C++ function), 148
bats::divide_conquer_flag (C++ struct), 39
bats::dowker_edge_param (C++ function), 148, 149
bats::dowker_filtration_edges (C++ function), 149
bats::DowkerFiltration (C++ function), 149, 150
bats::EilenbergZilber (C++ function), 150, 151
bats::enclosing_radius (C++ function), 151
bats::euclidean (C++ function), 151
bats::Euclidean (C++ struct), 39
bats::Euclidean::dist (C++ function), 39
bats::extension_perm (C++ function), 151
bats::extra_reduction_flag (C++ struct), 40
bats::extract_basis_indices (C++ function), 152
bats::extract_dimension (C++ function), 152
bats::extract_pairs (C++ function), 152
bats::filtered_edge (C++ struct), 40
bats::filtered_edge::filtered_edge (C++ function), 40
bats::filtered_edge::operator< (C++ function), 40
bats::filtered_edge::r (C++ member), 40
bats::filtered_edge::s (C++ member), 40
bats::filtered_edge::t (C++ member), 40
bats::FilteredChainComplex (C++ struct), 40
bats::FilteredChainComplex::C (C++ member), 41
bats::FilteredChainComplex::complex (C++ function), 41
bats::FilteredChainComplex::dim (C++ function), 41
bats::FilteredChainComplex::FilteredChainComplex (C++ function), 41
bats::FilteredChainComplex::perm (C++ member), 41
bats::FilteredChainComplex::update_filtration (C++ function), 41
bats::FilteredChainComplex::update_filtration_general (C++ function), 41
bats::FilteredChainComplex::val (C++ member), 41
bats::FilteredChainComplex::vals (C++ function), 41
bats::FilteredCochainComplex (C++ struct), 41
bats::FilteredCochainComplex::C (C++ member), 42
bats::FilteredCochainComplex::complex (C++ function), 41
bats::FilteredCochainComplex::dim (C++ function), 41
bats::FilteredCochainComplex::FilteredCochainComplex (C++ function), 41
bats::FilteredCochainComplex::perm (C++ member), 42

```

bats::FilteredCochainComplex::update_filtration (C++ function), 42
bats::FilteredCochainComplex::update_filtration (C++ function), 42
bats::FilteredCochainComplex::val (C++ member), 42
bats::FilteredCochainComplex::vals (C++ function), 42
bats::FilteredDGVectorSpace (C++ struct), 42
bats::FilteredDGVectorSpace::_set_perms (C++ function), 42
bats::FilteredDGVectorSpace::C (C++ member), 43
bats::FilteredDGVectorSpace::complex (C++ function), 42
bats::FilteredDGVectorSpace::dim (C++ function), 42
bats::FilteredDGVectorSpace::FilteredDGVectorSpace (C++ function), 42
bats::FilteredDGVectorSpace::perm (C++ member), 43
bats::FilteredDGVectorSpace::update_filtration (C++ function), 43
bats::FilteredDGVectorSpace::val (C++ member), 43
bats::FilteredDGVectorSpace::vals (C++ function), 42
bats::Filtration (C++ class), 82
bats::Filtration::add (C++ function), 83
bats::Filtration::add_directed (C++ function), 83
bats::Filtration::add_recursive (C++ function), 83
bats::Filtration::complex (C++ function), 83
bats::Filtration::Filtration (C++ function), 82, 83
bats::Filtration::maxdim (C++ function), 83
bats::Filtration::ncells (C++ function), 83
bats::Filtration::sortperm (C++ function), 83, 84
bats::Filtration::sublevelset (C++ function), 84
bats::Filtration::union_add (C++ function), 83
bats::Filtration::update_filtration (C++ function), 83
bats::Filtration::vals (C++ function), 83
bats::filtration_iperm (C++ function), 153
bats::filtration_sortperm (C++ function), 153
bats::find_parent (C++ function), 153
bats::find_perm_from_vector (C++ function), 154
bats::flag_filtration_edges (C++ function), 154
bats::FlagComplex (C++ function), 154
bats::FlagFiltration (C++ function), 154
bats::FlagFiltration_extension (C++ function), 155
bats::flags::divide_conquer (C++ struct), 43
bats::flags::leftward (C++ struct), 43
bats::flags::rightward (C++ struct), 43
bats::generalize_repel_rp (C++ function), 155
bats::Freudenthal (C++ function), 155–157
bats::future::ColumnMajor (C++ struct), 44
bats::future::CompositePermutation (C++ class), 84
bats::future::CompositePermutation::append (C++ function), 84
bats::future::CompositePermutation::CompositePermutation (C++ function), 84
bats::future::CompositePermutation::operator() (C++ function), 84
bats::future::CompositePermutation::operator<< (C++ function), 84
bats::future::CompositePermutation::swap (C++ function), 84
bats::future::const_strided_iterator (C++ class), 85
bats::future::const_strided_iterator::~const_strided_iterator (C++ function), 85
bats::future::const_strided_iterator::const_strided_iterator (C++ function), 85
bats::future::const_strided_iterator::operator!= (C++ function), 85
bats::future::const_strided_iterator::operator* (C++ function), 85
bats::future::const_strided_iterator::operator+ (C++ function), 85
bats::future::const_strided_iterator::operator++ (C++ function), 85
bats::future::const_strided_iterator::operator== (C++ function), 85
bats::future::const_strided_iterator::operator-- (C++ function), 85
bats::future::const_strided_iterator::operator-> (C++ function), 85
bats::future::const_strided_iterator::operator[] (C++ function), 85
bats::future::ElementaryPermutation (C++ struct), 44
bats::future::ElementaryPermutation::ElementaryPermutation (C++ function), 44
bats::future::ElementaryPermutation::i (C++ member), 44
bats::future::ElementaryPermutation::j (C++ member), 44
bats::future::ElementaryPermutation::operator() (C++ function), 44
bats::future::ElementaryPermutation::operator<< (C++ function), 45
bats::future::find_pivot_complete (C++ function), 157
bats::future::find_pivot_high (C++ function),

```


- 157, 158
- `bats::future::gemm (C++ function)`, 158
- `bats::future::gemv (C++ function)`, 158
- `bats::future::l_residual (C++ function)`, 159
- `bats::future::l_solve (C++ function)`, 159
- `bats::future::LU (C++ function)`, 160
- `bats::future::LUFact (C++ struct)`, 45
- `bats::future::LUFact::L (C++ member)`, 45
- `bats::future::LUFact::LUFact (C++ function)`, 45
- `bats::future::LUFact::P (C++ member)`, 45
- `bats::future::LUFact::print (C++ function)`, 45
- `bats::future::LUFact::print_info (C++ function)`, 45
- `bats::future::LUFact::prod (C++ function)`, 45
- `bats::future::LUFact::Q (C++ member)`, 45
- `bats::future::LUFact::U (C++ member)`, 45
- `bats::future::LUFact::val_type (C++ type)`, 45
- `bats::future::Matrix (C++ class)`, 86
- `bats::future::Matrix::add_column (C++ function)`, 88
- `bats::future::Matrix::add_row (C++ function)`, 87, 88
- `bats::future::Matrix::append_column (C++ function)`, 87
- `bats::future::Matrix::column (C++ function)`, 87
- `bats::future::Matrix::data (C++ function)`, 86
- `bats::future::Matrix::delete_column (C++ function)`, 87
- `bats::future::Matrix::dims (C++ function)`, 86
- `bats::future::Matrix::fill (C++ function)`, 86
- `bats::future::Matrix::identity (C++ function)`, 88
- `bats::future::Matrix::len (C++ function)`, 86
- `bats::future::Matrix::Matrix (C++ function)`, 86
- `bats::future::Matrix::mm (C++ function)`, 88
- `bats::future::Matrix::mv (C++ function)`, 88
- `bats::future::Matrix::ncol (C++ function)`, 86
- `bats::future::Matrix::nrow (C++ function)`, 86
- `bats::future::Matrix::operator() (C++ function)`, 87
- `bats::future::Matrix::operator* (C++ function)`, 88
- `bats::future::Matrix::operator[] (C++ function)`, 87
- `bats::future::Matrix::print (C++ function)`, 87
- `bats::future::Matrix::print_info (C++ function)`, 87
- `bats::future::Matrix::row (C++ function)`, 87
- `bats::future::Matrix::scale_column (C++ function)`, 88
- `bats::future::Matrix::scale_row (C++ function)`, 88
- `bats::future::Matrix::swap_columns (C++ function)`, 87
- `bats::future::Matrix::swap_rows (C++ function)`, 87
- `bats::future::Matrix::transpose (C++ function)`, 87
- `bats::future::Matrix::value_type (C++ type)`, 86
- `bats::future::Matrix::view (C++ function)`, 87
- `bats::future::MatrixView (C++ class)`, 88
- `bats::future::MatrixView::MatrixView (C++ function)`, 89
- `bats::future::MatrixView::ncol (C++ function)`, 89
- `bats::future::MatrixView::nrow (C++ function)`, 89
- `bats::future::MatrixView::operator() (C++ function)`, 89
- `bats::future::MatrixView::print (C++ function)`, 89
- `bats::future::MatrixView::print_info (C++ function)`, 89
- `bats::future::MatrixView::val_type (C++ type)`, 89
- `bats::future::operator<< (C++ function)`, 160
- `bats::future::Permutation (C++ class)`, 89
- `bats::future::Permutation::Permutation (C++ function)`, 89
- `bats::future::range (C++ class)`, 90
- `bats::future::range::begin (C++ function)`, 90
- `bats::future::range::cbegin (C++ function)`, 90
- `bats::future::range::cend (C++ function)`, 90
- `bats::future::range::const_iterator (C++ class)`, 91, 92
- `bats::future::range::const_iterator::~~const_iterator (C++ function)`, 91, 92
- `bats::future::range::const_iterator::const_iterator (C++ function)`, 91, 92
- `bats::future::range::const_iterator::operator!= (C++ function)`, 91, 92
- `bats::future::range::const_iterator::operator* (C++ function)`, 91, 92
- `bats::future::range::const_iterator::operator++ (C++ function)`, 91, 92
- `bats::future::range::const_iterator::operator== (C++ function)`, 91, 92
- `bats::future::range::const_iterator::operator-- (C++ function)`, 91, 92
- `bats::future::range::const_iterator::operator-> (C++ function)`, 91, 92
- `bats::future::range::end (C++ function)`, 90
- `bats::future::range::first (C++ function)`, 90
- `bats::future::range::iterator (C++ type)`, 90
- `bats::future::range::last (C++ function)`, 90
- `bats::future::range::operator[] (C++ function)`, 90
- `bats::future::range::range (C++ function)`, 90

bats::future::range::size (C++ function), 90
 bats::future::range::stride (C++ function), 90
 bats::future::RowMajor (C++ struct), 46
 bats::future::SimilarityTransform (C++ struct), 46
 bats::future::SimilarityTransform::A (C++ member), 47
 bats::future::SimilarityTransform::add_row (C++ function), 46
 bats::future::SimilarityTransform::operator() (C++ function), 46
 bats::future::SimilarityTransform::print (C++ function), 46
 bats::future::SimilarityTransform::print_info (C++ function), 46
 bats::future::SimilarityTransform::prod (C++ function), 46
 bats::future::SimilarityTransform::S (C++ member), 47
 bats::future::SimilarityTransform::scale_column (C++ function), 46
 bats::future::SimilarityTransform::scale_row (C++ function), 46
 bats::future::SimilarityTransform::SimilarityTransform (C++ function), 46
 bats::future::SimilarityTransform::Sinv (C++ member), 47
 bats::future::SimilarityTransform::size (C++ function), 46
 bats::future::SimilarityTransform::swap_columns (C++ function), 46
 bats::future::SimilarityTransform::swap_rows (C++ function), 46
 bats::future::Span (C++ class), 93
 bats::future::Span::_vdim (C++ member), 93
 bats::future::Span::add (C++ function), 93
 bats::future::Span::contains (C++ function), 93
 bats::future::Span::dim (C++ function), 93
 bats::future::Span::L (C++ member), 93
 bats::future::Span::Pt (C++ member), 93
 bats::future::Span::Span (C++ function), 93
 bats::future::Span::vdim (C++ function), 93
 bats::future::strided_iterator (C++ class), 94
 bats::future::strided_iterator::~~strided_iterator (C++ function), 94
 bats::future::strided_iterator::operator!= (C++ function), 94
 bats::future::strided_iterator::operator* (C++ function), 94
 bats::future::strided_iterator::operator+ (C++ function), 94
 bats::future::strided_iterator::operator++ (C++ function), 94
 bats::future::strided_iterator::operator== (C++ function), 94
 bats::future::strided_iterator::operator-- (C++ function), 94
 bats::future::strided_iterator::operator-> (C++ function), 94
 bats::future::strided_iterator::operator[] (C++ function), 94
 bats::future::strided_iterator::strided_iterator (C++ function), 94
 bats::future::unit_lower (C++ function), 160
 bats::future::upper (C++ function), 161
 bats::future::VectorView (C++ class), 95
 bats::future::VectorView::axpy (C++ function), 95
 bats::future::VectorView::begin (C++ function), 95
 bats::future::VectorView::cbegin (C++ function), 95
 bats::future::VectorView::cend (C++ function), 95
 bats::future::VectorView::const_iterator (C++ type), 95
 bats::future::VectorView::end (C++ function), 95
 bats::future::VectorView::iterator (C++ type), 95
 bats::future::VectorView::operator[] (C++ function), 95
 bats::future::VectorView::print (C++ function), 95
 bats::future::VectorView::scale (C++ function), 95
 bats::future::VectorView::VectorView (C++ function), 95
 bats::gen_cylinder (C++ function), 161
 bats::get_clearing_inds (C++ function), 161
 bats::get_compression_inds (C++ function), 161
 bats::get_dM_ZZ_inds (C++ function), 162
 bats::get_m (C++ function), 162
 bats::get_subset (C++ function), 162
 bats::greedy_landmarks (C++ function), 162
 bats::greedy_landmarks_hausdorff (C++ function), 163
 bats::hausdorff_landmarks (C++ function), 164
 bats::Hom (C++ function), 164, 165
 bats::identity_perm (C++ function), 165
 bats::increment_m (C++ function), 166
 bats::induced_map (C++ function), 166, 167
 bats::interval (C++ function), 167
 bats::is_left_arrow (C++ function), 167
 bats::k_choose_2 (C++ function), 168
 bats::k_choose_2_inv (C++ function), 168
 bats::kdist (C++ function), 168
 bats::Kendall_tau (C++ function), 168, 169
 bats::kron_chain_shift (C++ function), 170

bats::kron_chains (C++ function), 170
 bats::kron_homology (C++ function), 170
 bats::kron_index (C++ function), 171
 bats::L1Dist (C++ struct), 47
 bats::L1Dist::dist (C++ function), 47
 bats::landmark_cover (C++ function), 171
 bats::landmark_eps_cover (C++ function), 171
 bats::LightSimplicialComplex (C++ class), 96
 bats::LightSimplicialComplex::add (C++ function), 97
 bats::LightSimplicialComplex::add_recursive (C++ function), 97
 bats::LightSimplicialComplex::add_unsafe (C++ function), 97
 bats::LightSimplicialComplex::boundary (C++ function), 97
 bats::LightSimplicialComplex::boundary_csc (C++ function), 97
 bats::LightSimplicialComplex::find_idx (C++ function), 97
 bats::LightSimplicialComplex::get_cell (C++ function), 96
 bats::LightSimplicialComplex::get_simplex (C++ function), 96
 bats::LightSimplicialComplex::get_simplices (C++ function), 96, 97
 bats::LightSimplicialComplex::key_to_simplex (C++ function), 96
 bats::LightSimplicialComplex::LightSimplicialComplexMatrix (C++ type), 246
 bats::LightSimplicialComplex::max_vertex (C++ function), 96
 bats::LightSimplicialComplex::maxdim (C++ function), 96
 bats::LightSimplicialComplex::ncells (C++ function), 96
 bats::LightSimplicialComplex::print_summary (C++ function), 97
 bats::LightSimplicialComplex::simplex_begin (C++ function), 97
 bats::LightSimplicialComplex::simplex_boundary_iterator (C++ struct), 47
 bats::LightSimplicialComplex::simplex_boundary_iterator::bool (C++ function), 48
 bats::LightSimplicialComplex::simplex_boundary_iterator::C (C++ function), 48
 bats::LightSimplicialComplex::simplex_boundary_iterator::C (C++ function), 48
 bats::LightSimplicialComplex::simplex_boundary_iterator::C (C++ function), 48
 bats::LightSimplicialComplex::simplex_boundary_iterator::C (C++ function), 48
 bats::LightSimplicialComplex::simplex_boundary_iterator::p (C++ member), 48
 bats::LightSimplicialComplex::simplex_boundary_iterator::s (C++ function), 48
 bats::LightSimplicialComplex::simplex_end (C++ function), 97
 bats::LightSimplicialComplex::simplex_key (C++ function), 96
 bats::linear_cover_intersection_diagram (C++ function), 171
 bats::linear_subset_union_diagram (C++ function), 172
 bats::LInfDist (C++ struct), 49
 bats::LInfDist::dist (C++ function), 49
 bats::LowerStarFiltration (C++ function), 172
 bats::make_edge (C++ function), 172
 bats::mayer_vietoris_boundary (C++ function), 172
 bats::neighborhood (C++ function), 173
 bats::neighborhoods (C++ function), 173, 174
 bats::Nerve (C++ function), 174
 bats::no_apparent_pairs_flag (C++ struct), 49
 bats::NO_IND (C++ member), 244
 bats::no_optimization_flag (C++ struct), 49
 bats::normalize_entries (C++ function), 175
 bats::operator< (C++ function), 175
 bats::operator<< (C++ function), 175
 bats::oscillatingRipsZigzagSets (C++ function), 175
 bats::pairwise_dist (C++ function), 176
 bats::partial_reduce_parallel (C++ function), 176
 bats::before (C++ function), 177
 bats::pass_L_left (C++ function), 177
 bats::pass_R_left (C++ function), 177
 bats::pass_P_right (C++ function), 177
 bats::pass_P_left (C++ function), 177
 bats::pass_U_right (C++ function), 178
 bats::pass_UP_right (C++ function), 178
 bats::perm_to_the_end (C++ function), 178
 bats::PersistencePair (C++ struct), 50
 bats::PersistencePair::birth (C++ member), 50

```

bats::PersistencePair::birth_ind (C++ member), 50
bats::PersistencePair::death (C++ member), 50
bats::PersistencePair::death_ind (C++ member), 50
bats::PersistencePair::dim (C++ member), 50
bats::PersistencePair::get_birth (C++ function), 50
bats::PersistencePair::get_birth_ind (C++ function), 50
bats::PersistencePair::get_death (C++ function), 50
bats::PersistencePair::get_death_ind (C++ function), 50
bats::PersistencePair::get_dim (C++ function), 50
bats::PersistencePair::length (C++ function), 50
bats::PersistencePair::mid (C++ function), 50
bats::PersistencePair::operator!= (C++ function), 50
bats::PersistencePair::operator== (C++ function), 50
bats::PersistencePair::PersistencePair (C++ function), 50
bats::PersistencePair::str (C++ function), 50
bats::print_1D_vectors (C++ function), 179
bats::print_2D_vectors (C++ function), 179
bats::print_filtration_info (C++ function), 179
bats::print_simplex (C++ function), 179
bats::print_summary_of_filtration (C++ function), 180
bats::prod_ind (C++ function), 180
bats::product_paths (C++ function), 180
bats::product_space (C++ function), 181
bats::random_landmarks (C++ function), 181
bats::read_point_cloud (C++ function), 181
bats::Reduce (C++ function), 181, 182
bats::reduce_block_dq (C++ function), 182, 183
bats::reduce_block_sequential (C++ function), 184
bats::reduce_column_extra (C++ function), 185
bats::reduce_column_sparsify (C++ function), 185
bats::reduce_column_standard (C++ function), 186
bats::reduce_matrix (C++ function), 187, 188
bats::reduce_matrix_clearing (C++ function), 189
bats::reduce_matrix_compression (C++ function), 189, 190
bats::reduce_matrix_extra (C++ function), 190
bats::reduce_matrix_standard (C++ function), 190, 191
bats::ReducedChainComplex (C++ class), 97
bats::ReducedChainComplex::betti (C++ function), 98
bats::ReducedChainComplex::chain_preferred_representative (C++ function), 99
bats::ReducedChainComplex::chain_type (C++ type), 98
bats::ReducedChainComplex::dim (C++ function), 98
bats::ReducedChainComplex::find_preferred_representative (C++ function), 99
bats::ReducedChainComplex::from_hom_basis (C++ function), 99
bats::ReducedChainComplex::get_preferred_representative (C++ function), 99
bats::ReducedChainComplex::hdim (C++ function), 98
bats::ReducedChainComplex::I (C++ member), 99
bats::ReducedChainComplex::initialize (C++ function), 98
bats::ReducedChainComplex::maxdim (C++ function), 98
bats::ReducedChainComplex::operator[] (C++ function), 98
bats::ReducedChainComplex::p2c (C++ member), 99
bats::ReducedChainComplex::permute_basis (C++ function), 99
bats::ReducedChainComplex::permute_matrices (C++ function), 99
bats::ReducedChainComplex::print_summary (C++ function), 99
bats::ReducedChainComplex::R (C++ member), 99
bats::ReducedChainComplex::ReducedChainComplex (C++ function), 98
bats::ReducedChainComplex::remove_extra_cycles (C++ function), 99
bats::ReducedChainComplex::sparsify_basis (C++ function), 99
bats::ReducedChainComplex::to_hom_basis (C++ function), 99
bats::ReducedChainComplex::U (C++ member), 99
bats::ReducedChainComplex::update_basis_general (C++ function), 99
bats::ReducedChainComplex::update_reduction (C++ function), 98
bats::ReducedChainComplex::update_reduction2 (C++ function), 98
bats::ReducedCochainComplex (C++ class), 100
bats::ReducedCochainComplex::chain_preferred_representative (C++ function), 101
bats::ReducedCochainComplex::cochain_type (C++ type), 100
bats::ReducedCochainComplex::dim (C++ member), 101
bats::ReducedCochainComplex::find_preferred_representative (C++ function), 101
bats::ReducedCochainComplex::from_hom_basis (C++ function), 101

```

(C++ function), 100

bats::ReducedCochainComplex::get_preferred_representation (C++ function), 100

bats::ReducedCochainComplex::hdim (C++ function), 100

bats::ReducedCochainComplex::I (C++ member), 101

bats::ReducedCochainComplex::maxdim (C++ function), 100

bats::ReducedCochainComplex::operator[] (C++ function), 100

bats::ReducedCochainComplex::p2c (C++ member), 101

bats::ReducedCochainComplex::R (C++ member), 101

bats::ReducedCochainComplex::ReducedCochainComplex (C++ function), 100

bats::ReducedCochainComplex::to_hom_basis (C++ function), 100

bats::ReducedCochainComplex::U (C++ member), 101

bats::ReducedDGVectorSpace (C++ struct), 51

bats::ReducedDGVectorSpace::_delete_cells (C++ function), 52

bats::ReducedDGVectorSpace::_insert_cells (C++ function), 52

bats::ReducedDGVectorSpace::_make_U_upper_triangular (C++ function), 52

bats::ReducedDGVectorSpace::beti (C++ function), 51

bats::ReducedDGVectorSpace::chain_preferred_representation (C++ function), 52

bats::ReducedDGVectorSpace::degree (C++ member), 53

bats::ReducedDGVectorSpace::dim (C++ function), 51

bats::ReducedDGVectorSpace::find_preferred_representation (C++ function), 52

bats::ReducedDGVectorSpace::find_reverse_index (C++ function), 52

bats::ReducedDGVectorSpace::from_hom_basis (C++ function), 52

bats::ReducedDGVectorSpace::get_preferred_representation (C++ function), 52

bats::ReducedDGVectorSpace::hdim (C++ function), 51

bats::ReducedDGVectorSpace::I (C++ member), 53

bats::ReducedDGVectorSpace::initialize (C++ function), 51

bats::ReducedDGVectorSpace::maxdim (C++ function), 51

bats::ReducedDGVectorSpace::operator[] (C++ function), 51

bats::ReducedDGVectorSpace::p2c (C++ member), 53

bats::ReducedDGVectorSpace::permute_basis (C++ function), 52

bats::ReducedDGVectorSpace::permute_matrices (C++ function), 52

bats::ReducedDGVectorSpace::print_summary (C++ function), 52

bats::ReducedDGVectorSpace::R (C++ member), 53

bats::ReducedDGVectorSpace::ReducedDGVectorSpace (C++ function), 51, 52

bats::ReducedDGVectorSpace::set_indices (C++ function), 51

bats::ReducedDGVectorSpace::to_hom_basis (C++ function), 52

bats::ReducedDGVectorSpace::U (C++ member), 53

bats::ReducedDGVectorSpace::update_basis (C++ function), 52

bats::ReducedDGVectorSpace::update_basis_general (C++ function), 53

bats::ReducedDGVectorSpace::update_basis_general_clearing (C++ function), 53

bats::ReducedDGVectorSpace::update_reduction2 (C++ function), 52

bats::ReducedDGVectorSpace::vect_type (C++ type), 51

bats::ReducedFilteredChainComplex (C++ struct), 53

bats::ReducedFilteredChainComplex::barcode (C++ function), 54

bats::ReducedFilteredChainComplex::critical_cells (C++ function), 54

bats::ReducedFilteredChainComplex::dim (C++ function), 53

bats::ReducedFilteredChainComplex::get_nnz_R (C++ function), 54

bats::ReducedFilteredChainComplex::get_nnz_U (C++ function), 54

bats::ReducedFilteredChainComplex::get_subcomplex (C++ function), 54

bats::ReducedFilteredChainComplex::hdim (C++ function), 53

bats::ReducedFilteredChainComplex::maxdim (C++ function), 53

bats::ReducedFilteredChainComplex::perm (C++ member), 55

bats::ReducedFilteredChainComplex::persistence_pairs (C++ function), 53

bats::ReducedFilteredChainComplex::persistence_pairs_vec (C++ function), 54

bats::ReducedFilteredChainComplex::print_summary (C++ function), 54

bats::ReducedFilteredChainComplex::RC (C++ member), 55

bats::ReducedFilteredChainComplex::ReducedFilteredChainComplex

(C++ function), 53
 bats::ReducedFilteredChainComplex::remove_extra_cycles (C++ function), 54
 bats::ReducedFilteredChainComplex::representatives (C++ function), 54
 bats::ReducedFilteredChainComplex::sparsify_basis (C++ function), 54
 bats::ReducedFilteredChainComplex::update_filtration (C++ function), 54
 bats::ReducedFilteredChainComplex::update_filtration_singular (C++ function), 54
 bats::ReducedFilteredChainComplex::val (C++ member), 55
 bats::ReducedFilteredDGVectorSpace (C++ struct), 55
 bats::ReducedFilteredDGVectorSpace::dim (C++ function), 55
 bats::ReducedFilteredDGVectorSpace::hdim (C++ function), 55
 bats::ReducedFilteredDGVectorSpace::maxdim (C++ function), 55
 bats::ReducedFilteredDGVectorSpace::perm (C++ member), 56
 bats::ReducedFilteredDGVectorSpace::persistences_pairs (C++ function), 55
 bats::ReducedFilteredDGVectorSpace::persistences_pairs_singular (C++ function), 55
 bats::ReducedFilteredDGVectorSpace::RC (C++ member), 56
 bats::ReducedFilteredDGVectorSpace::ReducedFilteredDGVectorSpace (C++ function), 55
 bats::ReducedFilteredDGVectorSpace::update_basis (C++ function), 55
 bats::ReducedFilteredDGVectorSpace::update_filtration (C++ function), 55
 bats::ReducedFilteredDGVectorSpace::update_filtration_singular (C++ function), 55
 bats::ReducedFilteredDGVectorSpace::update_filtration_singular_deleting (C++ function), 55
 bats::ReducedFilteredDGVectorSpace::val (C++ member), 56
 bats::remove_extra_cycles (C++ function), 191
 bats::Rips (C++ function), 191, 192
 bats::rips_edges (C++ function), 192
 bats::rips_filtration_edges (C++ function), 193
 bats::rips_union_find_pairs (C++ function), 193
 bats::RipsComplex (C++ function), 194
 bats::RipsFiltration (C++ function), 194, 195
 bats::RipsFiltration_extension (C++ function), 195
 bats::RipsHom (C++ function), 195
 bats::rowmajor::get_idx (C++ function), 196
 bats::RPAngleDist (C++ struct), 56
 bats::RPAngleDist::dist (C++ function), 56
 bats::RPCosineDist (C++ struct), 57
 bats::RPCosineDist::dist (C++ function), 57
 bats::sample_cube (C++ function), 196
 bats::sample_sphere (C++ function), 197
 bats::serpinski_diagram (C++ function), 197
 bats::SimplicialComplex (C++ class), 101
 bats::SimplicialComplex::~SimplicialComplex
 bats::SimplicialComplex::add (C++ function), 102
 bats::SimplicialComplex::add_directed (C++ function), 102
 bats::SimplicialComplex::add_directed_recursive (C++ function), 102
 bats::SimplicialComplex::add_recursive (C++ function), 102
 bats::SimplicialComplex::boundary (C++ function), 103
 bats::SimplicialComplex::boundary_csc (C++ function), 103
 bats::SimplicialComplex::faces_begin (C++ function), 102
 bats::SimplicialComplex::faces_end (C++ function), 102
 bats::SimplicialComplex::find_idx (C++ function), 101, 102
 bats::SimplicialComplex::get_cell (C++ function), 102
 bats::SimplicialComplex::get_indices (C++ function), 103
 bats::SimplicialComplex::get_simplex (C++ function), 102
 bats::SimplicialComplex::get_simplices (C++ function), 103
 bats::SimplicialComplex::intersection (C++ function), 103
 bats::SimplicialComplex::maxdim (C++ function), 102
 bats::SimplicialComplex::make_ncells (C++ function), 102
 bats::SimplicialComplex::print_cells (C++ function), 103
 bats::SimplicialComplex::print_summary (C++ function), 103
 bats::SimplicialComplex::save (C++ function), 103
 bats::SimplicialComplex::simplex_begin (C++ function), 102
 bats::SimplicialComplex::simplex_end (C++ function), 102
 bats::SimplicialComplex::simplicial_union (C++ function), 103
 bats::SimplicialComplex::SimplicialComplex (C++ function), 101
 bats::SimplicialComplex::union_add (C++ function), 102

- tion), 103
 bats::SimplicialMap (C++ function), 197, 198
 bats::sort_edges (C++ function), 198
 bats::sort_indexes (C++ function), 198
 bats::sparse_reduction_flag (C++ struct), 57
 bats::SparseTrie (C++ class), 104
 bats::SparseTrie::~~SparseTrie (C++ function), 104
 bats::SparseTrie::child_container (C++ type), 104
 bats::SparseTrie::child_type (C++ type), 104
 bats::SparseTrie::children (C++ member), 105
 bats::SparseTrie::count (C++ function), 105
 bats::SparseTrie::emplace (C++ function), 104
 bats::SparseTrie::get (C++ function), 104, 105
 bats::SparseTrie::insert (C++ function), 104
 bats::SparseTrie::operator= (C++ function), 104
 bats::SparseTrie::operator[] (C++ function), 104
 bats::SparseTrie::SparseTrie (C++ function), 104
 bats::SparseTrie::val (C++ member), 105
 bats::sparsify_basis (C++ function), 198, 199
 bats::standard_reduction_flag (C++ struct), 57
 bats::StrictRipsCoverFiltration (C++ function), 199
 bats::StrictWitnessComplex (C++ function), 199
 bats::tedge (C++ struct), 57
 bats::tedge::s (C++ member), 58
 bats::tedge::t (C++ member), 58
 bats::tedge::tedge (C++ function), 58
 bats::tedge::v (C++ member), 58
 bats::test_reduce_result (C++ function), 200
 bats::triangle (C++ struct), 58
 bats::triangle::a (C++ member), 58
 bats::triangle::ab (C++ member), 58
 bats::triangle::b (C++ member), 58
 bats::triangle::bc (C++ member), 58
 bats::triangle::c (C++ member), 58
 bats::triangle::ca (C++ member), 58
 bats::triangle::triangle (C++ function), 58
 bats::TriangulatedProduct (C++ function), 200
 bats::type_A_dq_common (C++ function), 201
 bats::type_A_dq_EL (C++ function), 201
 bats::type_A_dq_EU (C++ function), 201
 bats::type_A_leftright_sweep1 (C++ function), 201
 bats::type_A_leftright_sweep2 (C++ function), 202
 bats::type_A_rightleft_sweep1 (C++ function), 202
 bats::type_A_rightleft_sweep2 (C++ function), 202
 bats::uniform_interval_cover (C++ function), 202
 bats::union_find_pairs (C++ function), 203
 bats::Update_info (C++ struct), 59
 bats::Update_info::addition_indices (C++ member), 59
 bats::Update_info::boundary_indices (C++ member), 59
 bats::Update_info::degree (C++ member), 60
 bats::Update_info::deletion_indices (C++ member), 59
 bats::Update_info::F_new (C++ member), 60
 bats::Update_info::F_old (C++ member), 60
 bats::Update_info::F_X_perms (C++ member), 59
 bats::Update_info::F_X_vals (C++ member), 59
 bats::Update_info::F_Y_perms (C++ member), 59
 bats::Update_info::F_Y_vals (C++ member), 59
 bats::Update_info::filtered_boolean (C++ member), 60
 bats::Update_info::filtered_info (C++ function), 59
 bats::Update_info::intersection_indices_X (C++ member), 59
 bats::Update_info::intersection_indices_Y (C++ member), 59
 bats::Update_info::kendall_tau_dists (C++ member), 59
 bats::Update_info::max_dim (C++ member), 59
 bats::Update_info::perms_X_inv (C++ member), 59
 bats::Update_info::perms_Y_inv (C++ member), 59
 bats::Update_info::permutation_deletion_end (C++ function), 59
 bats::Update_info::permutations (C++ member), 59
 bats::Update_info::print_detail (C++ function), 59
 bats::Update_info::print_summary (C++ function), 59
 bats::Update_info::Update_info (C++ function), 59
 bats::UpdateInfo2 (C++ struct), 60
 bats::UpdateInfo2::insertion_cols (C++ member), 60
 bats::UpdateInfo2::insertion_indices (C++ member), 60
 bats::UpdateInfo2::ndeletions (C++ member), 60
 bats::UpdateInfo2::newperm (C++ function), 60
 bats::UpdateInfo2::newval (C++ member), 60
 bats::UpdateInfo2::perm (C++ member), 60
 bats::UpdateInfo2::perms2 (C++ member), 60
 bats::UpdateInfo2::reverse_for_cohomology (C++ function), 60
 bats::UpdateInfo2::reversed (C++ member), 60
 bats::UpdateInfo2::UpdateInfo2 (C++ function), 60
 bats::util::apply_iperm_swap (C++ function), 203

bats::util::apply_perm (C++ function), 203, 204
bats::util::apply_perm_swap (C++ function), 204
bats::util::fill_partial_sortperm (C++ function), 204
bats::util::fill_sortperm (C++ function), 204, 205
bats::util::find_sorted_lt (C++ function), 205
bats::util::firstk (C++ function), 205
bats::util::has_intersect_sorted (C++ function), 205
bats::util::has_intersect_sorted_lt (C++ function), 206
bats::util::intersect_sorted (C++ function), 206
bats::util::intersect_sorted_lt (C++ function), 206
bats::util::inv_perm (C++ function), 207
bats::util::io::parse_argv (C++ function), 207
bats::util::is_degenerate (C++ function), 207
bats::util::partial_perm (C++ function), 208
bats::util::perm_inversions (C++ function), 208
bats::util::perm_sign (C++ function), 208
bats::util::rand_perm (C++ function), 208
bats::util::random_subset (C++ function), 209
bats::util::read_simplex (C++ function), 209
bats::util::set_intersection (C++ function), 209
bats::util::set_union (C++ function), 209
bats::util::simplex_sign (C++ function), 210
bats::util::SimplexContainer (C++ class), 105
bats::util::SimplexContainer::dim (C++ function), 105
bats::util::SimplexContainer::emplace_back (C++ function), 105
bats::util::SimplexContainer::SimplexContainer (C++ function), 105
bats::util::SimplexContainer::size (C++ function), 105
bats::util::SimplexHasher (C++ struct), 61
bats::util::SimplexHasher::operator() (C++ function), 61
bats::util::sort_ind_by_perm (C++ function), 210
bats::util::sort_ind_pair_by_perm (C++ function), 210
bats::util::sort_into (C++ function), 210
bats::util::sort_sum_reduce (C++ function), 211
bats::util::sorted_complement (C++ function), 211
bats::util::sortperm (C++ function), 211, 212
bats::util::stable_sortperm (C++ function), 212
bats::util::to_set (C++ function), 213
bats::util::top_k (C++ function), 213
bats::util::top_p (C++ function), 213
bats::util::write_simplex (C++ function), 213, 214
bats::vertex_inclusion_map (C++ function), 214
bats::witness_edge_param (C++ function), 214
bats::witness_edges (C++ function), 214, 215
bats::witness_neighborhoods (C++ function), 215
bats::WitnessComplex (C++ function), 215, 216
bats::WitnessFiltration (C++ function), 216
bats::zigzag::barcode (C++ function), 216
bats::zigzag::detail::apply_basis (C++ function), 216
bats::zigzag::detail::boundary_insertion_map (C++ function), 217
bats::zigzag::detail::cube_extrema (C++ function), 217
bats::zigzag::detail::cube_val (C++ function), 217
bats::zigzag::detail::cycle_insertion_map (C++ function), 218
bats::zigzag::detail::lex_cmp (C++ function), 218
bats::zigzag::detail::rlex_cmp (C++ function), 218
bats::zigzag::detail::simplex_extrema (C++ function), 219
bats::zigzag::detail::updateBars (C++ function), 219
bats::zigzag::extend_levelset (C++ function), 219
bats::zigzag::extend_zigzag_filtration (C++ function), 220
bats::zigzag::extra_col_reduction (C++ function), 220
bats::zigzag::prepare_ChainComplex (C++ function), 221
bats::zigzag::reduce_column (C++ function), 221, 222
bats::zigzag::rfilt_val (C++ struct), 61
bats::zigzag::rfilt_val::cind (C++ member), 61
bats::zigzag::rfilt_val::dim (C++ member), 61
bats::zigzag::rfilt_val::entry (C++ member), 61
bats::zigzag::rfilt_val::ind (C++ member), 61
bats::zigzag::rfilt_val::operator<< (C++ function), 62
bats::zigzag::rfilt_val::rfilt_val (C++ function), 61
bats::zigzag::rfilt_val::val (C++ member), 61
bats::zigzag::zigzag_barcode_reduction (C++ function), 222
bats::zigzag::ZigzagChainComplex (C++ struct), 62
bats::zigzag::ZigzagChainComplex::_correct_indices (C++ function), 62
bats::zigzag::ZigzagChainComplex::C (C++ member), 63
bats::zigzag::ZigzagChainComplex::cind (C++

member), 63
 bats::zigzag::ZigzagChainComplex::dim (C++ function), 62
 bats::zigzag::ZigzagChainComplex::maxdim (C++ function), 62
 bats::zigzag::ZigzagChainComplex::val (C++ member), 63
 bats::zigzag::ZigzagChainComplex::vals (C++ function), 62
 bats::zigzag::ZigzagChainComplex::ZigzagChainComplex (C++ function), 62
 bats::zigzag::ZigzagFiltration (C++ class), 106
 bats::zigzag::ZigzagFiltration::add (C++ function), 106, 107
 bats::zigzag::ZigzagFiltration::add_recursive (C++ function), 107
 bats::zigzag::ZigzagFiltration::complex (C++ function), 106
 bats::zigzag::ZigzagFiltration::levelset (C++ function), 107
 bats::zigzag::ZigzagFiltration::maxdim (C++ function), 106
 bats::zigzag::ZigzagFiltration::ncells (C++ function), 106
 bats::zigzag::ZigzagFiltration::vals (C++ function), 106
 bats::zigzag::ZigzagFiltration::ZigzagFiltration (C++ function), 106
 bats::zigzag::ZigzagPair (C++ struct), 63
 bats::zigzag::ZigzagPair::birth (C++ member), 64
 bats::zigzag::ZigzagPair::birth_ind (C++ member), 64
 bats::zigzag::ZigzagPair::birth_is_entry (C++ member), 64
 bats::zigzag::ZigzagPair::death (C++ member), 64
 bats::zigzag::ZigzagPair::death_ind (C++ member), 64
 bats::zigzag::ZigzagPair::death_is_entry (C++ member), 64
 bats::zigzag::ZigzagPair::dim (C++ member), 64
 bats::zigzag::ZigzagPair::get_birth (C++ function), 63
 bats::zigzag::ZigzagPair::get_birth_ind (C++ function), 63
 bats::zigzag::ZigzagPair::get_death (C++ function), 63
 bats::zigzag::ZigzagPair::get_death_ind (C++ function), 63
 bats::zigzag::ZigzagPair::get_dim (C++ function), 63
 bats::zigzag::ZigzagPair::length (C++ function), 63
 bats::zigzag::ZigzagPair::mid (C++ function), 63
 bats::zigzag::ZigzagPair::operator!= (C++ function), 63
 bats::zigzag::ZigzagPair::operator== (C++ function), 63
 bats::zigzag::ZigzagPair::operator<< (C++ function), 64
 bats::zigzag::ZigzagPair::str (C++ function), 63
 bats::zigzag::ZigzagPair::ZigzagPair (C++ function), 63
 bats::zigzag_levelsets (C++ function), 223
 bats::zigzag_toplex (C++ function), 223
 block_reduce (C++ function), 223

C

cell_ind (C++ struct), 64
 cell_ind::cell_ind (C++ function), 64
 cell_ind::dim (C++ member), 64
 cell_ind::ind (C++ member), 64
 cell_ind::str (C++ function), 64
 characteristic_matrix (C++ function), 224
 ColumnMatrix (C++ class), 107
 ColumnMatrix::add_rows (C++ function), 111
 ColumnMatrix::append_column (C++ function), 109
 ColumnMatrix::append_row (C++ function), 109
 ColumnMatrix::append_sparse_row (C++ function), 109
 ColumnMatrix::block (C++ function), 110
 ColumnMatrix::clear_cols (C++ function), 110
 ColumnMatrix::clear_rows (C++ function), 110
 ColumnMatrix::col_inv_scale (C++ function), 112
 ColumnMatrix::col_type (C++ type), 108
 ColumnMatrix::cols (C++ function), 108
 ColumnMatrix::ColumnMatrix (C++ function), 108
 ColumnMatrix::direct_sum (C++ function), 113
 ColumnMatrix::dump_dense (C++ function), 110
 ColumnMatrix::erase_column (C++ function), 109
 ColumnMatrix::erase_final_columns (C++ function), 109
 ColumnMatrix::erase_final_rows_unsafe (C++ function), 110
 ColumnMatrix::erase_initial_cols (C++ function), 110
 ColumnMatrix::erase_initial_rows (C++ function), 110
 ColumnMatrix::erase_row (C++ function), 110
 ColumnMatrix::erase_row_unsafe (C++ function), 110
 ColumnMatrix::erase_rows (C++ function), 110
 ColumnMatrix::gemv (C++ function), 111
 ColumnMatrix::getval (C++ function), 108
 ColumnMatrix::identity (C++ function), 113
 ColumnMatrix::insert_column (C++ function), 109
 ColumnMatrix::insert_columns (C++ function), 109

- ColumnMatrix::insert_row (C++ function), 109
 ColumnMatrix::insert_rows (C++ function), 109
 ColumnMatrix::insert_sparse_rows (C++ function), 109
 ColumnMatrix::ipermute_cols (C++ function), 111
 ColumnMatrix::ipermute_rows (C++ function), 108
 ColumnMatrix::is_EL (C++ function), 112
 ColumnMatrix::is_ELhat (C++ function), 112
 ColumnMatrix::is_EU (C++ function), 112
 ColumnMatrix::is_EUhat (C++ function), 112
 ColumnMatrix::is_lower (C++ function), 112
 ColumnMatrix::is_pivot_matrix (C++ function), 112
 ColumnMatrix::is_reduced (C++ function), 112
 ColumnMatrix::is_upper (C++ function), 112
 ColumnMatrix::is_upper_invert (C++ function), 112
 ColumnMatrix::is_zero (C++ function), 112
 ColumnMatrix::J_conjugation (C++ function), 111
 ColumnMatrix::J_conjugation_inplace (C++ function), 111
 ColumnMatrix::J_left (C++ function), 111
 ColumnMatrix::J_left_inplace (C++ function), 111
 ColumnMatrix::J_right (C++ function), 111
 ColumnMatrix::J_right_inplace (C++ function), 111
 ColumnMatrix::mix_cols (C++ function), 111
 ColumnMatrix::mix_rows (C++ function), 110
 ColumnMatrix::ncol (C++ function), 108
 ColumnMatrix::nnz (C++ function), 112
 ColumnMatrix::nrow (C++ function), 108
 ColumnMatrix::operator() (C++ function), 110
 ColumnMatrix::operator* (C++ function), 111, 113
 ColumnMatrix::operator+ (C++ function), 111
 ColumnMatrix::operator+= (C++ function), 111
 ColumnMatrix::operator== (C++ function), 110
 ColumnMatrix::operator- (C++ function), 111
 ColumnMatrix::operator[] (C++ function), 110
 ColumnMatrix::permute (C++ function), 111
 ColumnMatrix::permute_cols (C++ function), 111
 ColumnMatrix::permute_rows (C++ function), 108
 ColumnMatrix::print (C++ function), 112
 ColumnMatrix::print_size (C++ function), 112
 ColumnMatrix::random (C++ function), 113
 ColumnMatrix::read (C++ function), 108
 ColumnMatrix::row_scale (C++ function), 112
 ColumnMatrix::save (C++ function), 112
 ColumnMatrix::schur_complement (C++ function), 112
 ColumnMatrix::set_block (C++ function), 110
 ColumnMatrix::set_nrow (C++ function), 108
 ColumnMatrix::str (C++ function), 112
 ColumnMatrix::submatrix (C++ function), 110
 ColumnMatrix::swap_cols (C++ function), 112
 ColumnMatrix::swap_rows (C++ function), 110
 ColumnMatrix::T (C++ function), 111
 ColumnMatrix::tensor_product (C++ function), 113
 ColumnMatrix::tmp_type (C++ type), 108
 ColumnMatrix::to_row_array (C++ function), 110
 ColumnMatrix::transpose (C++ function), 111
 ColumnMatrix::val_type (C++ type), 108
 ColumnMatrix::write (C++ function), 112
 commute (C++ function), 224
 complete_pairs (C++ function), 224
 CSCMatrix (C++ class), 113
 CSCMatrix::block_select (C++ function), 114
 CSCMatrix::CSCMatrix (C++ function), 113
 CSCMatrix::gemm (C++ function), 114
 CSCMatrix::get_colptr (C++ function), 114
 CSCMatrix::get_rowind (C++ function), 114
 CSCMatrix::get_val (C++ function), 114
 CSCMatrix::getval (C++ function), 113
 CSCMatrix::ncol (C++ function), 114
 CSCMatrix::nrow (C++ function), 114
 CSCMatrix::operator* (C++ function), 114
 CSCMatrix::print (C++ function), 114
 CSCMatrix::print_size (C++ function), 114
 CSCMatrix::submatrix (C++ function), 114
 CSCMatrix::sum (C++ function), 114
 CSCMatrix::trilu (C++ function), 114
 CU_inplace (C++ function), 225
- ## D
- D (C++ struct), 65
 delete_pivot (C++ function), 225
- ## E
- E (C++ struct), 66
 EL (C++ struct), 66
 EL_L_commute (C++ function), 225
 ELH (C++ struct), 66
 EU (C++ struct), 67
 EU_U_commute (C++ function), 225
 EUH (C++ struct), 67
 extend_filtration (C++ function), 226
 extended_euclidean (C++ function), 226
 extract_row_scale (C++ function), 226
- ## F
- ff_inv (C++ function), 226
- ## G
- gemv (C++ function), 227
 generating_basis (C++ function), 227
 get_pivots (C++ function), 227
- ## I
- IMPL_COUT (C macro), 245

- IMPL_FACT (*C macro*), 245
 inv (*C++ function*), 227
 is_UnivariatePolynomial (*C++ struct*), 68
 is_UnivariatePolynomial<UnivariatePolynomial<*T*>> (*C++ struct*), 68
 isprime (*C++ function*), 228
- ## L
- L (*C++ struct*), 69
 L_EL_commute (*C++ function*), 228
 l_inv (*C++ function*), 228
 l_solve (*C++ function*), 228, 229
 LEUP (*C++ function*), 229
 LEUP_Fact (*C++ function*), 229
 LEUP_fact (*C++ function*), 229
 LEUP_inplace (*C++ function*), 230
 lower_star_backwards (*C++ function*), 230
 lower_star_filtration (*C++ function*), 230, 231
 LQU (*C++ function*), 231
 LQU_inplace (*C++ function*), 231
- ## M
- MAT (*C++ struct*), 69
 matmul (*C++ function*), 232, 233
 max_cube_val (*C++ function*), 233
 MM (*C macro*), 245
 ModP (*C++ class*), 115
 ModP::inv (*C++ function*), 116
 ModP::iszero (*C++ function*), 116
 ModP::ModP (*C++ function*), 115
 ModP::operator!= (*C++ function*), 115
 ModP::operator* (*C++ function*), 115
 ModP::operator*= (*C++ function*), 115
 ModP::operator+ (*C++ function*), 115
 ModP::operator+= (*C++ function*), 115
 ModP::operator/ (*C++ function*), 116
 ModP::operator/= (*C++ function*), 116
 ModP::operator== (*C++ function*), 115
 ModP::operator- (*C++ function*), 115
 ModP::operator-= (*C++ function*), 115
 ModP::operator< (*C++ function*), 116
 ModP::operator<< (*C++ function*), 116
 ModP::to_int (*C++ function*), 115
 ModP<IntT, 2> (*C++ class*), 116
 ModP<IntT, 2>::inv (*C++ function*), 117
 ModP<IntT, 2>::iszero (*C++ function*), 117
 ModP<IntT, 2>::ModP (*C++ function*), 116
 ModP<IntT, 2>::operator!= (*C++ function*), 117
 ModP<IntT, 2>::operator* (*C++ function*), 117
 ModP<IntT, 2>::operator*= (*C++ function*), 117
 ModP<IntT, 2>::operator+ (*C++ function*), 116
 ModP<IntT, 2>::operator+= (*C++ function*), 116
 ModP<IntT, 2>::operator/ (*C++ function*), 117
 ModP<IntT, 2>::operator/= (*C++ function*), 117
 ModP<IntT, 2>::operator= (*C++ function*), 117
 ModP<IntT, 2>::operator== (*C++ function*), 117
 ModP<IntT, 2>::operator- (*C++ function*), 117
 ModP<IntT, 2>::operator-= (*C++ function*), 117
 ModP<IntT, 2>::operator< (*C++ function*), 117
 ModP<IntT, 2>::operator<< (*C++ function*), 117
 ModP<IntT, 2>::to_int (*C++ function*), 116
 MorsePairing (*C++ class*), 118
 MorsePairing::add (*C++ function*), 118
 MorsePairing::add_pair (*C++ function*), 119
 MorsePairing::boundary_csc (*C++ function*), 119
 MorsePairing::clear (*C++ function*), 118
 MorsePairing::down_paired (*C++ function*), 118
 MorsePairing::faces_begin (*C++ function*), 119
 MorsePairing::faces_end (*C++ function*), 119
 MorsePairing::is_paired (*C++ function*), 118
 MorsePairing::maxdim (*C++ function*), 118
 MorsePairing::MorsePairing (*C++ function*), 118
 MorsePairing::ncells (*C++ function*), 118
 MorsePairing::set_pair (*C++ function*), 118
 MorsePairing::set_pair_edge (*C++ function*), 118
 MorsePairing::size (*C++ function*), 118
 MorsePairing::unpaired (*C++ function*), 118
 MorsePairing::up_paired (*C++ function*), 118
 MultiGraph (*C++ class*), 119
 MultiGraph::add_edge (*C++ function*), 119
 MultiGraph::add_node (*C++ function*), 119
 MultiGraph::Edge (*C++ struct*), 70, 119
 MultiGraph::Edge::data (*C++ member*), 70, 120
 MultiGraph::Edge::Edge (*C++ function*), 70, 120
 MultiGraph::Edge::get_data (*C++ function*), 70, 120
 MultiGraph::Edge::print (*C++ function*), 70, 120
 MultiGraph::Edge::src (*C++ member*), 70, 120
 MultiGraph::Edge::targ (*C++ member*), 70, 120
 MultiGraph::MultiGraph (*C++ function*), 119
 MultiGraph::Node (*C++ struct*), 70, 120
 MultiGraph::Node::add_input (*C++ function*), 70, 120
 MultiGraph::Node::add_output (*C++ function*), 70, 120
 MultiGraph::Node::data (*C++ member*), 71, 120
 MultiGraph::Node::get_data (*C++ function*), 70, 120
 MultiGraph::Node::input (*C++ member*), 71, 120
 MultiGraph::Node::Node (*C++ function*), 70, 120
 MultiGraph::Node::output (*C++ member*), 71, 120
 MultiGraph::Node::print (*C++ function*), 71, 120
- ## N
- nzpair (*C++ struct*), 71
 nzpair::ind (*C++ member*), 72
 nzpair::nzpair (*C++ function*), 71
 nzpair::operator!= (*C++ function*), 71

`nzpair::operator==` (C++ function), 71
`nzpair::val` (C++ member), 72

O

`operator<` (C++ function), 234
`operator<<` (C++ function), 234–236

P

`P` (C++ struct), 72
`p2c_type` (C macro), 245
`pivot_ind` (C++ function), 236
`PLEU` (C++ function), 237
`PLEU_Fact` (C++ function), 237
`PLEU_fact` (C++ function), 237
`PLEU_inplace` (C++ function), 237
`PUEL` (C++ function), 238
`PUEL_Fact` (C++ function), 238
`PUEL_fact` (C++ function), 238
`PUEL_inplace` (C++ function), 238

R

`Rational` (C++ class), 121
`Rational::inv` (C++ function), 122
`Rational::iszero` (C++ function), 122
`Rational::operator!=` (C++ function), 121, 122
`Rational::operator*` (C++ function), 121
`Rational::operator*=` (C++ function), 121
`Rational::operator+` (C++ function), 121
`Rational::operator+=` (C++ function), 121
`Rational::operator/` (C++ function), 122
`Rational::operator/=` (C++ function), 122
`Rational::operator==` (C++ function), 121
`Rational::operator-` (C++ function), 121
`Rational::operator-=` (C++ function), 121
`Rational::operator<` (C++ function), 122
`Rational::operator<<` (C++ function), 122
`Rational::Rational` (C++ function), 121
`Rational::to_int` (C++ function), 121
`RCF_basis` (C++ function), 239

S

`schur` (C++ function), 239, 240
`SetVector` (C++ class), 122
`SetVector::axpy` (C++ function), 123
`SetVector::get` (C++ function), 123
`SetVector::lastnz` (C++ function), 123
`SetVector::lower_bound` (C++ function), 123
`SetVector::nnz` (C++ function), 123
`SetVector::nzbegin` (C++ function), 122, 123
`SetVector::nzend` (C++ function), 122, 123
`SetVector::operator==` (C++ function), 124
`SetVector::permute` (C++ function), 123
`SetVector::print` (C++ function), 123

`SetVector::print_row` (C++ function), 123
`SetVector::replace` (C++ function), 123
`SetVector::set` (C++ function), 123
`SetVector::SetVector` (C++ function), 122
`SetVector::upper_bound` (C++ function), 123
`sgn` (C++ function), 240
`SI` (C++ struct), 72
`smith_factorization` (C++ function), 240
`smith_normal_form` (C++ function), 241
`smith_rows` (C++ function), 241
`SmithFact` (C++ struct), 72
`SmithFact::C` (C++ member), 73
`SmithFact::prod` (C++ function), 73
`SmithFact::R` (C++ member), 73
`SmithFact::S` (C++ member), 73
`SparseFact` (C++ struct), 73
`SparseFact::E` (C++ member), 73
`SparseFact::L` (C++ member), 73
`SparseFact::LEUP_prod` (C++ function), 73
`SparseFact::LQU_prod` (C++ function), 73
`SparseFact::P` (C++ member), 73
`SparseFact::PLEU_prod` (C++ function), 73
`SparseFact::PUEL_prod` (C++ function), 73
`SparseFact::U` (C++ member), 73
`SparseFact::UELP_prod` (C++ function), 73
`SparseFact::UQL_prod` (C++ function), 73
`SparseVector` (C++ class), 124
`SparseVector::axpy` (C++ function), 128
`SparseVector::block` (C++ function), 126
`SparseVector::cast_values` (C++ function), 124
`SparseVector::clear` (C++ function), 125
`SparseVector::clear_dealloc` (C++ function), 125
`SparseVector::clear_inds` (C++ function), 125
`SparseVector::clear_zeros` (C++ function), 125
`SparseVector::coeff_intersection` (C++ function), 127
`SparseVector::dense` (C++ function), 129
`SparseVector::emplace_back` (C++ function), 126
`SparseVector::erase_all` (C++ function), 127
`SparseVector::erase_for_matrix` (C++ function), 127
`SparseVector::erase_initial_rows` (C++ function), 127
`SparseVector::erase_last_row_of_matrix` (C++ function), 127
`SparseVector::erase_last_rows_of_matrix` (C++ function), 127
`SparseVector::erase_rows_after` (C++ function), 127
`SparseVector::find_last_nz` (C++ function), 126
`SparseVector::firstnz` (C++ function), 126
`SparseVector::getval` (C++ function), 126
`SparseVector::insert_rows` (C++ function), 127
`SparseVector::ipermute` (C++ function), 126

- SparseVector::J (C++ function), 127
 SparseVector::kron (C++ function), 128
 SparseVector::lastnz (C++ function), 126
 SparseVector::lower_bound (C++ function), 126
 SparseVector::mix_rows (C++ function), 127
 SparseVector::nnz (C++ function), 125
 SparseVector::nnz_intersection (C++ function), 127
 SparseVector::nzbegin (C++ function), 125
 SparseVector::nzend (C++ function), 125
 SparseVector::nzinds (C++ function), 125
 SparseVector::nzs (C++ function), 126
 SparseVector::nzvals (C++ function), 125
 SparseVector::operator!= (C++ function), 125
 SparseVector::operator* (C++ function), 128
 SparseVector::operator+ (C++ function), 128
 SparseVector::operator== (C++ function), 125
 SparseVector::operator[] (C++ function), 125, 126
 SparseVector::permute (C++ function), 126
 SparseVector::print (C++ function), 128
 SparseVector::print_row (C++ function), 128
 SparseVector::random (C++ function), 129
 SparseVector::replace (C++ function), 126
 SparseVector::reverse_inds (C++ function), 124
 SparseVector::scale (C++ function), 128
 SparseVector::scale_inplace (C++ function), 127, 128
 SparseVector::set_block (C++ function), 126
 SparseVector::shift_inds (C++ function), 125
 SparseVector::sort (C++ function), 124
 SparseVector::SparseVector (C++ function), 124, 125
 SparseVector::str (C++ function), 128
 SparseVector::subvector (C++ function), 126
 SparseVector::swap_rows (C++ function), 127
 SparseVector::tensor (C++ function), 128
 SparseVector::tmp_type (C++ type), 124
 SparseVector::upper_bound (C++ function), 126
 SparseVector::val_type (C++ type), 124
 SparseVector::write (C++ function), 128
 standard_reduce (C++ function), 241
- T**
- T (C++ struct), 74
- U**
- U (C++ struct), 75
 U_EU_commute (C++ function), 241
 u_inv (C++ function), 242
 u_solve (C++ function), 242
 UELP (C++ function), 242
 UELP_Fact (C++ function), 243
 UELP_fact (C++ function), 243
 UELP_inplace (C++ function), 243
 UnivariatePolynomial (C++ class), 129
 UnivariatePolynomial::companion_matrix (C++ function), 130
 UnivariatePolynomial::degree (C++ function), 129
 UnivariatePolynomial::dim (C++ function), 129
 UnivariatePolynomial::divrem (C++ function), 130
 UnivariatePolynomial::gcd (C++ function), 130
 UnivariatePolynomial::identity (C++ function), 131
 UnivariatePolynomial::is_monic (C++ function), 130
 UnivariatePolynomial::is_zero (C++ function), 129
 UnivariatePolynomial::leading_coeff (C++ function), 129
 UnivariatePolynomial::monomial (C++ function), 131
 UnivariatePolynomial::operator!= (C++ function), 130
 UnivariatePolynomial::operator() (C++ function), 129, 130
 UnivariatePolynomial::operator* (C++ function), 130
 UnivariatePolynomial::operator+ (C++ function), 130
 UnivariatePolynomial::operator+= (C++ function), 130
 UnivariatePolynomial::operator/ (C++ function), 130
 UnivariatePolynomial::operator== (C++ function), 130
 UnivariatePolynomial::operator- (C++ function), 130
 UnivariatePolynomial::operator-= (C++ function), 130
 UnivariatePolynomial::operator<< (C++ function), 131
 UnivariatePolynomial::operator[] (C++ function), 129
 UnivariatePolynomial::print (C++ function), 130
 UnivariatePolynomial::remainder (C++ function), 130
 UnivariatePolynomial::size (C++ function), 129
 UnivariatePolynomial::UnivariatePolynomial (C++ function), 129
 UnivariatePolynomial::zero (C++ function), 131
 update_pivot (C++ function), 243
 UQL (C++ function), 244