NAME

FingerprintsBitVector

SYNOPSIS

use Fingerprints::FingerprintsBitVector;

use Fingerprints::FingerprintsBitVector qw(:coefficients);

use Fingerprints::FingerprintsBitVector qw(:all);

DESCRIPTION

FingerprintsBitVector class provides the following methods:

new, BaroniUrbaniSimilarityCoefficient, BuserSimilarityCoefficient, CosineSimilarityCoefficient, DennisSimilarityCoefficient, DiceSimilarityCoefficient, FoldFingerprintsBitVectorByDensity, FoldFingerprintsBitVectorBySize, ForbesSimilarityCoefficient, FossumSimilarityCoefficient, GetBitsAsBinaryString, GetBitsAsDecimalString, GetBitsAsHexadecimalString, GetBitsAsOctalString, GetBitsAsRawBinaryString, GetDescription, GetFingerprintsBitDensity, GetID, GetSpecifiedSize, GetSupportedSimilarityCoefficients, GetVectorType, HamannSimilarityCoefficient, IsFingerprintsBitVector, IsSubSet, JacardSimilarityCoefficient, Kulczynski1SimilarityCoefficient, Kulczynski2SimilarityCoefficient, MatchingSimilarityCoefficient, McConnaugheySimilarityCoefficient, NewFromBinaryString, NewFromDecimalString, NewFromHexadecimalString, NewFromOctalString, NewFromRawBinaryString, OchiaiSimilarityCoefficient, PearsonSimilarityCoefficient, RogersTanimotoSimilarityCoefficient, RussellRaoSimilarityCoefficient, SetDescription, SetID, SetSpecifiedSize, SetVectorType, SimpsonSimilarityCoefficient, SkoalSneath1SimilarityCoefficient, SkoalSneath2SimilarityCoefficient, SkoalSneath3SimilarityCoefficient, StringifyFingerprintsBitVector, TanimotoSimilarityCoefficient, TverskySimilarityCoefficient, WeightedTanimotoSimilarityCoefficient, WeightedTverskySimilarityCoefficient, YuleSimilarityCoefficient

The methods available to create fingerprints bit vector from strings and to calculate similarity coefficient between two bit vectors can also be invoked as class functions.

FingerprintsBitVector class is derived from BitVector class which provides the functionality to manipulate bits.

For two fingerprints bit vectors A and B of same size, let:

```
Na = Number of bits set to "1" in A
Nb = Number of bits set to "1" in B
Nc = Number of bits set to "1" in both A and B
Nd = Number of bits set to "0" in both A and B

Nt = Number of bits set to "1" or "0" in A or B (Size of A or B)
Nt = Na + Nb - Nc + Nd

Na - Nc = Number of bits set to "1" in A but not in B
Nb - Nc = Number of bits set to "1" in B but not in A
```

Then, various similarity coefficients [Ref. 40 - 42] for a pair of bit vectors A and B are defined as follows:

BaroniUrbani: (SQRT(Nc * Nd) + Nc) / (SQRT (Nc * Nd) + Nc + (Na - Nc) + (Nb - Nc)) (same as Buser)

Buser: (SQRT (Nc * Nd) + Nc) / (SQRT (Nc * Nd) + Nc + (Na - Nc) + (Nb - Nc)) (same as BaroniUrbani)

Cosine: Nc / SQRT (Na * Nb) (same as Ochiai)

Dice: (2 * Nc) / (Na + Nb)

Dennis: (Nc * Nd - ((Na - Nc) * (Nb - Nc))) / SQRT (Nt * Na * Nb)

Forbes: (Nt * Nc) / (Na * Nb)

Fossum: (Nt * ((Nc - 1/2) ** 2) / (Na * Nb)

Hamann: ((Nc + Nd) - (Na - Nc) - (Nb - Nc)) / Nt

Jaccard: Nc / ((Na - Nc) + (Nb - Nc) + Nc) = Nc / (Na + Nb - Nc) (same as Tanimoto)

Kulczynski1: Nc / ((Na - Nc) + (Nb - Nc)) = Nc / (Na + Nb - 2Nc)

Kulczynski2: ((Nc / 2) * (2 * Nc + (Na - Nc) + (Nb - Nc))) / ((Nc + (Na - Nc)) * (Nc + (Nb - Nc))) = 0.5 * (Nc / Na + Nc / Nb)

Matching: (Nc + Nd) / Nt

McConnaughey: (Nc ** 2 - (Na - Nc) * (Nb - Nc)) / (Na * Nb)

Ochiai: Nc / SQRT (Na * Nb) (same as Cosine)

Pearson: ((Nc * Nd) - ((Na - Nc) * (Nb - Nc)) / SQRT (Na * Nb * (Na - Nc + Nd) * (Nb - Nc + Nd))

RogersTanimoto: (Nc + Nd) / ((Na - Nc) + (Nb - Nc) + Nt) = (Nc + Nd) / (Na + Nb - 2Nc + Nt)

RussellRao: Nc / Nt

Simpson: Nc / MIN (Na, Nb)

SkoalSneath1: Nc / (Nc + 2 * (Na - Nc) + 2 * (Nb - Nc)) = Nc / (2 * Na + 2 * Nb - 3 * Nc)

SkoalSneath2: (2 * Nc + 2 * Nd) / (Nc + Nd + Nt)

SkoalSneath3: (Nc + Nd) / ((Na - Nc) + (Nb - Nc)) = (Nc + Nd) / (Na + Nb - 2 * Nc)

Tanimoto: Nc / ((Na - Nc) + (Nb - Nc) + Nc) = Nc / (Na + Nb - Nc) (same as Jaccard)

Tversky: Nc / (alpha * (Na - Nc) + (1 - alpha) * (Nb - Nc) + Nc) = Nc / (alpha * (Na - Nb) + Nb)

Yule: ((Nc * Nd) - ((Na - Nc) * (Nb - Nc))) / ((Nc * Nd) + ((Na - Nc) * (Nb - Nc)))

The values of Tanimoto/Jaccard and Tversky coefficients are dependent on only those bit which are set to "1" in both A and B. In order to take into account all bit positions, modified versions of Tanimoto [Ref. 42] and Tversky [Ref. 43] have been developed.

Let:

```
Na' = Number of bits set to "0" in A
Nb' = Number of bits set to "0" in B
Nc' = Number of bits set to "0" in both A and B
```

Tanimoto': Nc' / ((Na' - Nc') + (Nb' - Nc') + Nc') = Nc' / (Na' + Nb' - Nc')

Tversky': Nc' / (alpha * (Na' - Nc') + (1 - alpha) * (Nb' - Nc') + Nc') = Nc' / (alpha * (Na' - Nb') + Nb')

Then:

WeightedTanimoto = beta * Tanimoto + (1 - beta) * Tanimoto'

WeightedTversky = beta * Tversky + (1 - beta) * Tversky'

METHODS

new

```
$NewFPBitVector = new Fingerprints::FingerprintsBitVector($Size);
```

Creates a new *FingerprintsBitVector* object of size *Size* and returns newly created FingerprintsBitVector . Bit numbers range from 0 to 1 less than *Size*.

BaroniUrbaniSimilarityCoefficient

```
$Value = $FingerprintsBitVector->BaroniUrbaniSimilarityCoefficient(
          $OtherFingerprintBitVector);
$Value = Fingerprints::FingerprintsBitVector::
          BaroniUrbaniSimilarityCoefficient(
          $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *BaroniUrbani* similarity coefficient between two same size *FingerprintsBitVectors*.

BuserSimilarityCoefficient

```
$Value = $FingerprintsBitVector->BuserSimilarityCoefficient(
          $OtherFingerprintBitVector);
$Value = Fingerprints::FingerprintsBitVector::BuserSimilarityCoefficient(
```

```
                $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *Buser* similarity coefficient between two same size *FingerprintsBitVectors*.

CosineSimilarityCoefficient

```
    $Value = $FingerprintsBitVector->CosineSimilarityCoefficient(
                $OtherFingerprintBitVector);
    $Value = Fingerprints::FingerprintsBitVector::CosineSimilarityCoefficient(
                $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *Cosine* similarity coefficient between two same size *FingerprintsBitVectors*.

DennisSimilarityCoefficient

```
    $Value = $FingerprintsBitVector->DennisSimilarityCoefficient(
                $OtherFingerprintBitVector);
    $Value = Fingerprints::FingerprintsBitVector::DennisSimilarityCoefficient(
                $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *Dennis* similarity coefficient between two same size *FingerprintsBitVectors*.

DiceSimilarityCoefficient

```
    $Value = $FingerprintsBitVector->DiceSimilarityCoefficient(
                $OtherFingerprintBitVector);
    $Value = Fingerprints::FingerprintsBitVector::DiceSimilarityCoefficient(
                $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *Dice* similarity coefficient between two same size *FingerprintsBitVectors*.

FoldFingerprintsBitVectorByDensity

```
    $FingerprintsBitVector->FoldFingerprintsBitVectorByDensity($Density);
```

Folds *FingerprintsBitVector* by recursively reducing its size by half until bit density of set bits is greater than or equal to specified *Density* and returns folded *FingerprintsBitVector*.

FoldFingerprintsBitVectorBySize

```
    $FingerprintsBitVector->FoldFingerprintsBitVectorBySize($Size);
```

Folds *FingerprintsBitVector* by recursively reducing its size by half until size is less than or equal to specified *Size* and returns folded *FingerprintsBitVector*.

ForbesSimilarityCoefficient

```
    $Value = $FingerprintsBitVector->ForbesSimilarityCoefficient(
                $OtherFingerprintBitVector);
    $Value = Fingerprints::FingerprintsBitVector::ForbesSimilarityCoefficient(
                $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *Forbes* similarity coefficient between two same size *FingerprintsBitVectors*.

FossumSimilarityCoefficient

```
    $Value = $FingerprintsBitVector->FossumSimilarityCoefficient(
                $OtherFingerprintBitVector);
    $Value = Fingerprints::FingerprintsBitVector::FossumSimilarityCoefficient(
                $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *Fossum* similarity coefficient between two same size *FingerprintsBitVectors*.

GetBitsAsBinaryString

```
    $BinaryASCIIString = $FingerprintsBitVector->GetBitsAsBinaryString();
```

Returns fingerprints as a binary ASCII string containing 0s and 1s.

GetBitsAsHexadecimalString

```
    $HexadecimalString = $FingerprintsBitVector->GetBitsAsHexadecimalString();
```

Returns fingerprints as a hexadecimal string.

GetBitsAsRawBinaryString

```
$RawBinaryString = $FingerprintsBitVector->GetBitsAsRawBinaryString();
```

Returns fingerprints as a raw binary string containing packed bit values for each byte.

GetDescription

```
$Description = $FingerprintsBitVector->GetDescription();
```

Returns a string containing description of fingerprints bit vector.

GetFingerprintsBitDensity

```
$BitDensity = $FingerprintsBitVector->GetFingerprintsBitDensity();
```

Returns *BitDensity* of *FingerprintsBitVector* corresponding to bits set to 1s.

GetID

```
$ID = $FingerprintsBitVector->GetID();
```

Returns *ID* of *FingerprintsBitVector*.

GetVectorType

```
$VectorType = $FingerprintsBitVector->GetVectorType();
```

Returns *VectorType* of *FingerprintsBitVector*.

GetSpecifiedSize

```
$Size = $FingerprintsBitVector->GetSpecifiedSize();
```

Returns value of specified size for bit vector.

GetSupportedSimilarityCoefficients

```
@SimilarityCoefficient =
   Fingerprints::FingerprintsBitVector::GetSupportedSimilarityCoefficients();
```

Returns an array containing names of supported similarity coefficients.

HamannSimilarityCoefficient

```
$Value = $FingerprintsBitVector->HamannSimilarityCoefficient(
          $OtherFingerprintBitVector);
$Value = Fingerprints::FingerprintsBitVector::HamannSimilarityCoefficient(
          $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *Hamann* similarity coefficient between two same size *FingerprintsBitVectors*.

IsFingerprintsBitVector

```
$Status = Fingerprints::FingerprintsBitVector::
          IsFingerprintsBitVector($Object);
```

Returns 1 or 0 based on whether *Object* is a FingerprintsBitVector object.

IsSubSet

```
$Status = $FingerprintsBitVector->IsSubSet($OtherFPBitVector);
$Status = Fingerprints::FingerprintsBitVector::IsSubSet(
          $FPBitVectorA, $FPBitVectorB);
```

Returns 1 or 0 based on whether first firngerprints bit vector is a subset of second fingerprints bit vector.

For a bit vector to be a subset of another bit vector, both vectors must be of the same size and the bit positions set in first vector must also be set in the second bit vector.

JacardSimilarityCoefficient

```
$Value = $FingerprintsBitVector->JacardSimilarityCoefficient(
          $OtherFingerprintBitVector);
$Value = Fingerprints::FingerprintsBitVector::JacardSimilarityCoefficient(
          $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *Jacard* similarity coefficient between two same size *FingerprintsBitVectors*.

Kulczynski1SimilarityCoefficient

```
$Value = $FingerprintsBitVector->Kulczynski1SimilarityCoefficient(
          $OtherFingerprintBitVector);
$Value = Fingerprints::FingerprintsBitVector::
          Kulczynski1SimilarityCoefficient(
          $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *Kulczynski1* similarity coefficient between two same size *FingerprintsBitVectors*.

Kulczynski2SimilarityCoefficient

```
$Value = $FingerprintsBitVector->Kulczynski2SimilarityCoefficient(
          $OtherFingerprintBitVector);
$Value = Fingerprints::FingerprintsBitVector::
          Kulczynski2SimilarityCoefficient(
          $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *Kulczynski2* similarity coefficient between two same size *FingerprintsBitVectors*.

MatchingSimilarityCoefficient

```
$Value = $FingerprintsBitVector->MatchingSimilarityCoefficient(
          $OtherFingerprintBitVector);
$Value = Fingerprints::FingerprintsBitVector::
          MatchingSimilarityCoefficient(
          $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *Matching* similarity coefficient between two same size *FingerprintsBitVectors*.

McConnaugheySimilarityCoefficient

```
$Value = $FingerprintsBitVector->McConnaugheySimilarityCoefficient(
          $OtherFingerprintBitVector);
$Value = Fingerprints::FingerprintsBitVector::
          McConnaugheySimilarityCoefficient(
          $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *McConnaughey* similarity coefficient between two same size *FingerprintsBitVectors*.

NewFromBinaryString

```
$NewFPBitVector = $FingerprintsBitVector->NewFromBinaryString(
                    $BinaryString);
$NewFPBitVector = Fingerprints::FingerprintsBitVector::NewFromBinaryString(
                    $BinaryString);
```

Creates a new *FingerprintsBitVector* using *BinaryString* and returns new FingerprintsBitVector object.

NewFromHexadecimalString

```
$NewFPBitVector = $FingerprintsBitVector->NewFromHexadecimalString(
                    $HexdecimalString);
$NewFPBitVector = Fingerprints::FingerprintsBitVector::
                  NewFromHexadecimalString(
                    $HexdecimalString);
```

Creates a new *FingerprintsBitVector* using *HexdecimalString* and returns new FingerprintsBitVector object.

NewFromRawBinaryString

```
$NewFPBitVector = $FingerprintsBitVector->NewFromRawBinaryString(
                    $RawBinaryString);
$NewFPBitVector = Fingerprints::FingerprintsBitVector::
                    NewFromRawBinaryString(
                    $RawBinaryString);
```

Creates a new *FingerprintsBitVector* using *RawBinaryString* and returns new FingerprintsBitVector object.

OchiaiSimilarityCoefficient

```
$Value = $FingerprintsBitVector->OchiaiSimilarityCoefficient(
          $OtherFingerprintBitVector);
$Value = Fingerprints::FingerprintsBitVector::OchiaiSimilarityCoefficient(
```

```
                $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *Ochiai* similarity coefficient between two same size *FingerprintsBitVectors*.

PearsonSimilarityCoefficient

```
$Value = $FingerprintsBitVector->PearsonSimilarityCoefficient(
           $OtherFingerprintBitVector);
$Value = Fingerprints::FingerprintsBitVector::PearsonSimilarityCoefficient(
           $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *Pearson* similarity coefficient between two same size *FingerprintsBitVectors*.

RogersTanimotoSimilarityCoefficient

```
$Value = $FingerprintsBitVector->RogersTanimotoSimilarityCoefficient(
           $OtherFingerprintBitVector);
$Value = Fingerprints::FingerprintsBitVector::
           RogersTanimotoSimilarityCoefficient(
           $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *RogersTanimoto* similarity coefficient between two same size *FingerprintsBitVectors*.

RussellRaoSimilarityCoefficient

```
$Value = $FingerprintsBitVector->RussellRaoSimilarityCoefficient(
           $OtherFingerprintBitVector);
$Value = Fingerprints::FingerprintsBitVector::
           RussellRaoSimilarityCoefficient(
           $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *RussellRao* similarity coefficient between two same size *FingerprintsBitVectors*.

SetSpecifiedSize

```
$FingerprintsBitVector->SetSpecifiedSize($Size);
```

Sets specified size for fingerprints bit vector.

Irrespective of specified size, Perl functions used to handle bit data in BitVector class automatically sets the size to the next nearest power of 2. *SpecifiedSize* is used by FingerprintsBitVector class to process any aribitray size during similarity coefficient calculations.

SetDescription

```
$FingerprintsBitVector->SetDescription($Description);
```

Sets *Description* of fingerprints bit vector and returns *FingerprintsBitVector*.

SetID

```
$FingerprintsBitVector->SetID($ID);
```

Sets *ID* of fingerprints bit vector and returns *FingerprintsBitVector*.

SetVectorType

```
$FingerprintsBitVector->SetVectorType($VectorType);
```

Sets *VectorType* of fingerprints bit vector and returns *FingerprintsBitVector*.

SimpsonSimilarityCoefficient

```
$Value = $FingerprintsBitVector->SimpsonSimilarityCoefficient(
           $OtherFingerprintBitVector);
$Value = Fingerprints::FingerprintsBitVector::SimpsonSimilarityCoefficient(
           $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *Simpson* similarity coefficient between two same size *FingerprintsBitVectors*.

SkoalSneath1SimilarityCoefficient

```
$Value = $FingerprintsBitVector->SkoalSneath1SimilarityCoefficient(
           $OtherFingerprintBitVector);
$Value = Fingerprints::FingerprintsBitVector::
           SkoalSneath1SimilarityCoefficient(
```

```
                $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *SkoalSneath1* similarity coefficient between two same size *FingerprintsBitVectors*.

SkoalSneath2SimilarityCoefficient

```
$Value = $FingerprintsBitVector->SkoalSneath2SimilarityCoefficient(
          $OtherFingerprintBitVector);
$Value = Fingerprints::FingerprintsBitVector::
          SkoalSneath2SimilarityCoefficient(
          $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *SkoalSneath2* similarity coefficient between two same size *FingerprintsBitVectors*.

SkoalSneath3SimilarityCoefficient

```
$Value = $FingerprintsBitVector->SkoalSneath3SimilarityCoefficient(
          $OtherFingerprintBitVector);
$Value = Fingerprints::FingerprintsBitVector::
          SkoalSneath3SimilarityCoefficient(
          $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *SkoalSneath3* similarity coefficient between two same size *FingerprintsBitVectors*

StringifyFingerprintsBitVector

```
$String = $FingerprintsBitVector->StringifyFingerprintsBitVector();
```

Returns a string containing information about *FingerprintsBitVector* object.

TanimotoSimilarityCoefficient

```
$Value = $FingerprintsBitVector->TanimotoSimilarityCoefficient(
          $OtherFingerprintBitVector);
$Value = Fingerprints::FingerprintsBitVector::
          TanimotoSimilarityCoefficient(
          $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *Tanimoto* similarity coefficient between two same size *FingerprintsBitVectors*.

TverskySimilarityCoefficient

```
$Value = $FingerprintsBitVector->TverskySimilarityCoefficient(
          $OtherFingerprintBitVector, $Alpha);
$Value = Fingerprints::FingerprintsBitVector::
          TverskySimilarityCoefficient(
          $FingerprintsBitVectorA, $FingerprintBitVectorB, $Alpha);
```

Returns value of *Tversky* similarity coefficient between two same size *FingerprintsBitVectors*.

WeightedTanimotoSimilarityCoefficient

```
$Value =
   $FingerprintsBitVector->WeightedTanimotoSimilarityCoefficient(
     $OtherFingerprintBitVector, $Beta);
$Value =
   Fingerprints::FingerprintsBitVector::
     WeightedTanimotoSimilarityCoefficient(
     $FingerprintsBitVectorA, $FingerprintBitVectorB, $Beta);
```

Returns value of *WeightedTanimoto* similarity coefficient between two same size *FingerprintsBitVectors*.

WeightedTverskySimilarityCoefficient

```
$Value =
   $FingerprintsBitVector->WeightedTverskySimilarityCoefficient(
      $OtherFingerprintBitVector, $Alpha, $Beta);
$Value =
  Fingerprints::FingerprintsBitVector::
    WeightedTverskySimilarityCoefficient(
    $FingerprintsBitVectorA, $FingerprintBitVectorB, $Alpha, $Beta);
```

Returns value of *WeightedTversky* similarity coefficient between two same size *FingerprintsBitVectors*.

YuleSimilarityCoefficient

```
$Value = $FingerprintsBitVector->YuleSimilarityCoefficient(
          $OtherFingerprintBitVector);
$Value = Fingerprints::FingerprintsBitVector::YuleSimilarityCoefficient(
          $FingerprintsBitVectorA, $FingerprintBitVectorB);
```

Returns value of *Yule* similarity coefficient between two same size *FingerprintsBitVectors*.

AUTHOR

Manish Sud <msud@san.rr.com>

SEE ALSO

BitVector.pm, FingerprintsStringUtil.pm, FingerprintsVector.pm, Vector.pm

COPYRIGHT