

HCB

June 11, 2019

Title Human Cultural Boundaries

Version 0.0.0

Description Creates seed populations with phoneme inventories that can grow, migrate, and create off-shoot populations. Phoneme inventories mutate when populations establish a new territory.

License What license it uses

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1.9000

Imports mc2d, randomcoloR, uuid, numbers, philentropy, ade4

R topics documented:

AddBeringStrait	2
AddShift	2
AddSnakeBarriers	3
CardinalDirections	3
DefineParameters	4
GeneratePhonemeProbabilities	6
GenerateSeedLanguage	6
GetFactorDim	6
GetRealPhonemeData	7
HCBSimmulation	7
HoritontalTransferRepeater	7
HorizontalTransfer	8
Initialize	8
Lose	9
NextStepDirections	9
OneStepDirections	10
RemoveHorizontalConnections	10
RemoveVerticalConnections	11
ShiftDirections	11
StepDirections	12
Index	13

AddBeringStrait	<i>Add Bering Strait</i>
-----------------	--------------------------

Description

Removes connections at the FirstStep stage of the Local structure to create "barriers" between cells. Bering Strait Barriers are designed to create structures similar to the Bering strait entering North America, Traveling to Central America, then opening up into South America.

Usage

AddBeringStrait(P, firstStep)

Arguments

P	A list of parameters.
firstStep	The local directions created by OneStepDirections().

AddShift	<i>Add Shift Phoneme</i>
----------	--------------------------

Description

Allows a language to either gain a new phoneme and shift and existing phoneme to match another population.

Usage

AddShift(P, language, languages, local, phonemeRelatedness, index)

Arguments

P	A list of parameters.
language	The target language to be modified if possible.
languages	All languages
local	Locality.
index	The target territory whose language may change.
phonemeProbab	The probability of gaining each phoneme in the population.

AddSnakeBarriers	<i>Add Snake Barriers</i>
------------------	---------------------------

Description

Removes connections at the FirstStep stage of the Local structure to create "barriers" between cells. Snake Barriers are lines with length and spacing defined by the parameters. The barriers jut out from the east and west walls, alternating east, west, east, west. This creates a snaking zig-zag pattern, hence the name.

Usage

```
AddSnakeBarriers(P, firstStep)
```

Arguments

P	A list of parameters.
firstStep	The local directions created by OneStepDirections().

CardinalDirections	<i>Cardinal Directions</i>
--------------------	----------------------------

Description

Calculates the territory numbers of locations around a target territory (also used for phoneme relatedness in the same way).

Usage

```
CardinalDirections(target, R, start, round, South, North, East, West, SE,  
NE, SW, NW)
```

Arguments

target	The territory around which to get local territories.
R	The number of rows.
start	How much to offset numbers (for phoneme structures).
round	Whether to get a "round" set of territories (N, S, E, W only) for phonemes or a square set of territories (includes diagonals) for distance.
SE	Whether to get the southeastern territory.
NE	Whether to get the northeastern territory.
SW	Whether to get the southwestern territory.
NW	Whether to get the northwestern territory.
south	Whether to get the southern territory.
north	Whether to get the northern territory.
east	Whether to get the eastern territory.
west	Whether to get the western territory.

DefineParameters

*Define Parameters***Description**

Creates a parameter data structure for running simulations.

Usage

```
DefineParameters(Rows = 40, Cols = 50, ChanceExpand = 0.8,
  PopulationStartIndex = c(1, 2), NumPopulationPhonemes = rep(NA,
    length(PopulationStartIndex)), UsePopSize = TRUE,
  IndividualsStEmSuEM = c(1000, 10, 20, NA), MutationRate = 15,
  PhonemeDitribution = c(12, 24, 133), Consonants = 750,
  Vowels = 100, MinConsonant = 6, MinVowel = 6,
  PhonemeProbabilityType = "RealMimic", GrowthRate = 5,
  Barriers = FALSE, BarrierLength = 30, BarrierBreaks = 4,
  MutationTypeChance = rep(1/5, 5), HorizontalRate = 0.1,
  Bias = TRUE, Steps = 1, HorizontalLocal = TRUE,
  NumberRandomHorizontal = 8, UpRoot = TRUE, Death = TRUE,
  Bering = FALSE, MigrationSimSteps = 300, HorizontalSimSteps = 400,
  Waves = FALSE, Seed = NA)
```

Arguments

Rows	The number of rows in the world matrix.
Cols	The number of columns in the world matrix.
ChanceExpand	The chance that a population will either move or send off a group of individuals to found a new population.
PopulationStartIndex	The position in the matrix where each seed population starts. The number of seed populations is defined by the number of starting indicies.
NumPopulationPhonemes	The number of phonemes in each starting population. If set to NA, this is decided by sampling from a distribution with min, mode, and made on the values from the PhonemeDistribution arguement.
UsePopSize	Whether to take into account the the population size (number of people) when making decisions about moving, immigrating, and phoneme loss/addition biases.
IndividualsStEmSuEM	Four related parameters: 1) The number of individuals a seed population stats with, 2) the minumum number of individuals required to make a founder party to settle a new territory, 3) the minumum number of individuals that must stay behind when a founder party is sent off, 4) the maximum number of individuals allowed to be in one founder party.
MutationRate	The rate at which phonemes mutate. E.g., if MutationRate==0.1, each phoneme in a populatiosn phoneme inventory has a 10% chance to mutate.
Consonants	The number of possible consonants in existence. Default based on real phoneme data.

Vowels	The number of possible vowels in existence. Default based on real phoneme data.
MinConsonant	The minimum number of consonants that can be in a population's phoneme inventory. Default based on real phoneme data.
MinVowel	The minimum number of vowels that can be in a population's phoneme inventory. Default based on real phoneme data.
PhonemeProbabilityType	The method by which phoneme probabilities are established.
GrowthRate	When an integer, the number of individuals added to each population every time step. When a fraction, the percent that a population increases each timestep.
Barriers	Whether to create "snake barriers" that limit the direction of migration in the matrix.
BarrierLength	The width of snake barriers.
BarrierBreaks	The height of the space between snake barriers.
MutationTypeChance	The chance that each mutation type occurs. 1) Add, 2) Lose, 3) Split, 4) Join, 5) Shift.
HorizontalRate	The fraction of the population that attempts to modify its phoneme inventory every horizontal timestep.
Bias	Whether to randomly bias mutations towards either gains or losses when populations are small. Set to true based on previously published data.
Steps	The number of distance steps away from a target location that are considered "local." Includes all 8 cardinal and ordinal directions around a target, so the local area is always a rectangle around the target location.
HorizontalLocal	Whether horizontal transfer occurs between local populations or globally. Set to FALSE as a control, as global horizontal transfer should abolish local patterns.
NumberRandomHorizontal	The number of locations to compare when HorizontalLocal==FALSE. Should be 8 when Steps==1, 24 when steps==2, 48 when Steps=3, ect.
UpRoot	Whether established populations can move (TRUE) or they remain in place for the entire simulation (FALSE).
Death	Whether a population can die out.
Bering	Whether to employ barriers that mimic the Bering Strait and Americas.
MigrationSimSteps	The number of time steps to run each wave of migration.
HorizontalSimSteps	The number of time steps to spend on horizontal transfer.
Waves	Whether migration occurs in waves or all seed populations are added at the same time. If TRUE, there is one wave for each seed population.
Seed	Sets a seed for reproducibility if an integer instead of NA.
PhonemeDistribution	The 1) min, 2) mode, and 3) max number of phonemes a population can have when sampling for seed population sizes and when preventing languages from gaining or losing too many phonemes. Default based on real phoneme data.

GeneratePhonemeProbabilities

Generate Phoneme Probabilities

Description

Genetrate a vector of the probability to know each phoneme.

Usage

GeneratePhonemeProbabilities(P)

Arguments

P A list of parameters.

GenerateSeedLanguage *Generate Seed Language*

Description

Generate Seed Language

Usage

GenerateSeedLanguage(P, phonemeProbab, seedNum)

Arguments

P A list of parameters.
phonemeProbab The probability of gaining each phoneme in the population.
seedNum Which population seed is having it's language generated.

GetFactorDim *Get Factor Dimentions*

Description

Given a number of consomants or vowel, create a datastructure that is as square as possible.

Usage

GetFactorDim(nPhonemes)

Arguments

nPhonemes The number of Phonemes (vowels or consonants).

GetRealPhonemeData	<i>Get Real Phoneme Data</i>
--------------------	------------------------------

Description

Uses the real Phoneme data from Creanza..... UPDATE THIS!!!! to determine the phoneme probabilities.

Usage

```
GetRealPhonemeData(nPhoneme, actual, file)
```

Arguments

nPhoneme	The number of phonemes (vowels or consonants).
actual	Whether the data is Real (TRUE) or RealMimic (FALSE).

HCBSimulation	<i>Human CUltural Boundaries Simulation</i>
---------------	---

Description

Runs a simulation

Usage

```
HCBSimulation(P)
```

Arguments

P	A list of parameters.
---	-----------------------

HoritontalTransferRepeater	<i>After migration, allow populations to exchnage phoneme information, losing or gaining syllables based on other populations in the simulation.</i>
----------------------------	--

Description

After migration, allow populations to exchnage phoneme information, losing or gaining syllables based on other populations in the simulation.

Usage

```
HoritontalTransferRepeater(P, S)
```

Arguments

P	A list of parameters.
S	A list of the data structures.

HorizontalTransfer	<i>Horizontal Transfer</i>
--------------------	----------------------------

Description

A function wrapper that get the language to modify and allows the phoneme change to either add/shift or remove a phoneme if this can be done.

Usage

```
HorizontalTransfer(P, languages, local, phonemeRelatedness, phonemeProbab,
index)
```

Arguments

P	A list of parameters.
languages	All languages.
local	Locality.
phonemeRelatedness	The phoneme relatedness list.
phonemeProbab	The probability of gaining each phoneme in the population.
index	The target territory whose language may change.

Initialize	<i>Initialize</i>
------------	-------------------

Description

The function wrapper that makes calls to create the population and phoneme data structures and then populates them with initial data.

Usage

```
Initialize(P)
```

Arguments

P	A list of parameters.
---	-----------------------

Lose	<i>Lose Phoneme</i>
------	---------------------

Description

Allows a language to either lose a phoneme to better match other populations.

Usage

```
Lose(P, language, phonemeProbab)
```

Arguments

P	A list of parameters.
language	The target language to be modified if possible.
phonemeProbab	The probability of gaining each phoneme in the population.

NextStepDirections	<i>Next Step Directions Expands the Steps list one more step out.</i>
--------------------	---

Description

Next Step Directions Expands the Steps list one more step out.

Usage

```
NextStepDirections(firstStep, currentStep, start = 0)
```

Arguments

firstStep	The original StepOne.
currentStep	StepOne in its current state.
start	How much to offset numbers (for phoneme structures).

OneStepDirections	<i>One Step Directions</i>
-------------------	----------------------------

Description

One Step Directions

Usage

```
OneStepDirections(R, C, start = 0, round = FALSE)
```

Arguments

R	The number of rows.
C	The number of columns.
start	How much to offset numbers (for phoneme structures).
round	whether to make the spacing Round (Phonemes) or Square (Territories).

RemoveHorizontalConnections	<i>Remove Horizontal Connections</i>
-----------------------------	--------------------------------------

Description

Affects local territories below/South (and perhaps to the Southeast and Southwest) the target territory (index) and above/North (perhaps Northwest and Northeast) of index +1.

Usage

```
RemoveHorizontalConnections(R, index, firstStep, right = TRUE,  
  left = TRUE)
```

Arguments

R	The number of rows in the population matrix.
index	The target territory.
firstStep	The local directions created by OneStepDirections().
right	Whether to remove the right diagonal.
left	Whether to remove the left diagonal.

`RemoveVerticalConnections`*Remove Vertical Connections*

Description

Affects local territories right/East (and perhaps to the Northeast and Southeast) the target territory (index) and left/West (perhaps Northwest and Southwest) of index + R.

Usage

```
RemoveVerticalConnections(R, index, firstStep, above = TRUE,  
                          below = TRUE)
```

Arguments

R	The number of rows in the population matrix.
index	The target territory.
firstStep	The local directions created by OneStepDirections().
above	Whether to remove the upper diagonal.
below	Whether to remove the lower diagonal.

`ShiftDirections`*Shift Directions*

Description

Returns the relationships between phonemes with an offset of start.

Usage

```
ShiftDirections(nPhonemes, start = 0)
```

Arguments

nPhonemes	The number of phonemes.
start	Where to start number the phonemes (0 for consonants, number of consonants +1 for vowels).

StepDirections	<i>Step Directions</i>
----------------	------------------------

Description

A wrapper that calls StepOne(), add barriers if required, then expands StepOne as many steps as the Steps parameter calls for.

Usage

StepDirections(P)

Arguments

P	A list of parameters.
---	-----------------------

Index

*Topic **Barriers**

- AddBeringStrait, [2](#)
- AddSnakeBarriers, [3](#)
- RemoveHorizontalConnections, [10](#)
- RemoveVerticalConnections, [11](#)

*Topic **Directions**

- CardinalDirections, [3](#)
- GetFactorDim, [6](#)
- NextStepDirections, [9](#)
- OneStepDirections, [10](#)
- ShiftDirections, [11](#)
- StepDirections, [12](#)

*Topic **Horizontal**

- AddShift, [2](#)
- HoritontalTransferRepeater, [7](#)
- HorizontalTransfer, [8](#)
- Lose, [9](#)

*Topic **Phonemes**

- GeneratePhonemeProbabilities, [6](#)
- GenerateSeedLanguage, [6](#)
- GetRealPhonemeData, [7](#)

*Topic **SimParam**

- DefineParameters, [4](#)
- HCBSimulation, [7](#)

*Topic **Wrapper**

- Initialize, [8](#)
- StepDirections, [12](#)

Initialize, [8](#)

Lose, [9](#)

NextStepDirections, [9](#)

OneStepDirections, [10](#)

RemoveHorizontalConnections, [10](#)

RemoveVerticalConnections, [11](#)

ShiftDirections, [11](#)

StepDirections, [12](#)

AddBeringStrait, [2](#)

AddShift, [2](#)

AddSnakeBarriers, [3](#)

CardinalDirections, [3](#)

DefineParameters, [4](#)

GeneratePhonemeProbabilities, [6](#)

GenerateSeedLanguage, [6](#)

GetFactorDim, [6](#)

GetRealPhonemeData, [7](#)

HCBSimulation, [7](#)

HoritontalTransferRepeater, [7](#)

HorizontalTransfer, [8](#)