
SOFTWARE METAPAPER

Vowel System Sandbox: Complex System Modelling of Language Change

Sean A. Fulop¹ and Hannah Scott²

¹ Department of Linguistics at California State University Fresno, US

² School of Electrical Engineering and Computer Science at Oregon State University, US

Corresponding author: Sean A. Fulop (sfulop@mail.fresnostate.edu)

Vowel System Sandbox is a complex agent-based modelling tool which is intended for linguists and speech researchers to test hypotheses about how vowel sounds are transmitted and used through the generations in a language community, and thus how vowel systems may change over generational time. Written in Python 3, the code repository is on Github and can be run in Linux, Windows 7+ and MacOS. This is the first software that provides a computational model of sound change in language by implementing first principles of speech perception and production.

Keywords: language change; agent-based modelling; vowel systems; complex systems; Python

Funding statement: This project was partially funded by Provost awards for Research, Scholarship and Creative Activity at Fresno State University.

(1) Overview

Introduction

This paper describes software which applies agent-based modelling in a complex system to the problem of simulating language change in a speech community. In particular, the model focuses on the evolution of vowel sounds and sound systems in a community of agents transmitting word pronunciations to the “children” as the population goes through a number of life cycles. Since a great many parameters of the model are adjustable by the user, we call the software Vowel System Sandbox, or VoSS.

Language in general has often been characterized as a complex adaptive system [1, 2]. Unfortunately, nothing about language change has really been learned from this characterization because there is no agreed upon theory of the subject, few results have been derived from it, and moreover this idea has never been demonstrated in a fully complex model of a speech community. A number of advocates of the complex system viewpoint on language have proceeded to implement computational models, but these have often been based on very little actual knowledge of the transmission of language and speech and are thus overly simplistic, e.g. [3].

A complex (adaptive) system is definable as one in which a large number of individual objects interact according to locally governed parameters, which leads to global phenomena that emerge from the complexity without themselves being specifically parameterized [4]. A favourite example is that of a hurricane, which emerges from the local interactions of air molecules in the complex system of the atmosphere. VoSS was designed to demonstrate

how a population of agents (model speakers) whose local interactions are strictly governed can nevertheless show global patterns of change in the sounds spoken. The agent interaction mechanics are directly parameterized based on previous research into the first principles of speech production and perception [5], and speech transmission from person to person [6, 7]. As such, VoSS qualifies as a *microscopic* or *micro-level* complex system model, in that it directly implements agent-level interactions. We directly model vowel sounds only because they are relatively easy to represent in a realistic auditory space using the first two formant frequencies on auditory scales.

In linguistics, it seems that such a modelling tool for language change is highly desirable. Hamann has written that “a computer simulation that includes both phonetic and phonological changes by modelling the acquisition of phonetic and phonological categories where the speakers/listeners interact with several other agents does not exist yet [8]”. Recent related work is highly limited in comparison to VoSS. We leave aside efforts to model the genesis of language, including the genesis of vowel systems [9], since the problem addressed here is not the problem of how language as a communication system first emerged, but rather the problem of how established natural languages constantly change. There have been several papers which report modelling some specific aspect of language change between agents.

Harrison et al. [10] used the SWARM environment to model vowel harmony in a small group of agents. No acoustical representations were used, however, and vowels “change” in a symbolic sense according to probabilistic rules.

Clearly this fails to model the actual way in which vowels are transmitted, which is by sound production and perception.

Winter and Wedel [11] constructed a model with just two agents interacting. These agents “spoke” words to each other, the phonetic exemplars of which were represented by just two phonetic parameters on a 100-point scale. This sort of model is not a complex system model at all.

Chirkova and Gong [12] model specific vowel systems of one language, and their agents speak only vowels and not words. Their framework assumes that there is variation among speakers, and that adult listeners automatically try to incorporate all of the variation they hear and imitate it. This is but one example of the ubiquitous assumption that phoneme acquisition occurs by some kind of social synchronization, akin to that which influences the lights of some species of fireflies. This assumption appears to have no basis in reality, however.

More recently, Harrington and Schiel [13] model the single process of /u/-fronting in English. The proposed mechanism is incremental sound change due to mutual imitation. It must again be noted that there is no reason to accept that this mechanism obtains in real speech communities. It is not a mechanism that has been proven to cause sound change in fact. Moreover, this is a model of a single confined process, vowel fronting, to the exclusion of all other language factors. It is well-established in complex system modelling that this is always a terrible strategy.

Suffice to say that Hamann’s remark above is indeed correct, and that thanks to the recent advances in computational power available, VoSS represents the first effort we know of to fill the void with a model that is complex, represents a potentially large number of agents interacting, and which represents aspects of both the phonological and phonetic levels. With VoSS we have a genuine complex system that models vowel transmission at the micro level using a multitude of agents, represents vowels using their actual auditory parameters, and in which agents must acquire sounds when they are “babies” and words when they are “children”, later to “grow up” and stop learning sounds. The closest work in spirit is the iterated learning approach promoted by Kirby [14], but this has never been applied to the transmission of speech sounds. The virtual agents in VoSS transmit vowel pronunciations by a realistic process where the sound is articulated and the listening agent has to react to it at a cognitive level. Previous research has usually treated vowels as things that can be passed

around, but the reality of it is much more complicated and this requires a more sophisticated modelling effort. The macro-level effects on vowels are generally emergent from the system parameters, with a minimum of direct control in the model over the macro level.

Since VoSS is an agent-based model, one might duly wonder why we chose to write completely custom software in Python, rather than leveraging an existing agent-based modelling tool such as NetLogo [15]. A cursory examination makes it abundantly clear that VoSS could never be implemented in NetLogo because the VoSS agent interactions are more sophisticated than the kinds which can be simulated in that framework. NetLogo and other agent-based modelling platforms are intended for modelling systems with a small number of parameters where the outcomes can be suitably represented with simple charts. Moreover, NetLogo is intended for programming a microscale model but only observing the macroscale behaviour which results. It is not designed for directly observing the microscale behaviour, which is another important feature of VoSS.

Implementation and architecture

Structure of a simulation

The simulated speech community begins with a group of speakers/agents known as the *ancestors*. The ancestors are the initiators of the simulated language, which consists of a lexicon of one-syllable words that includes examples of all the vowel phonemes or *cogphones* in the particular simulation. At present all vowel phonemes are static, identified by one set of formant frequencies. There is no provision for diphthongs or vowel dynamics in this version. The words also contain a variety of consonants surrounding the vowels. All words have equal frequency of occurrence, but the sounds do not appear in an equal number of words. The ancestors “speak” words to transmit their language, but do not learn. The simulation is run from this initial point in a series of iterative “time steps” consisting of four main functions (see **Figure 1**).

1. Reproduce

A group of agents is added to the population. These *babies* acquire the vowel cogphones by listening to a set number of older agents (consistent across their lifespan) to build their vowel *repertoire*. The total population size is limited by a user parameter.

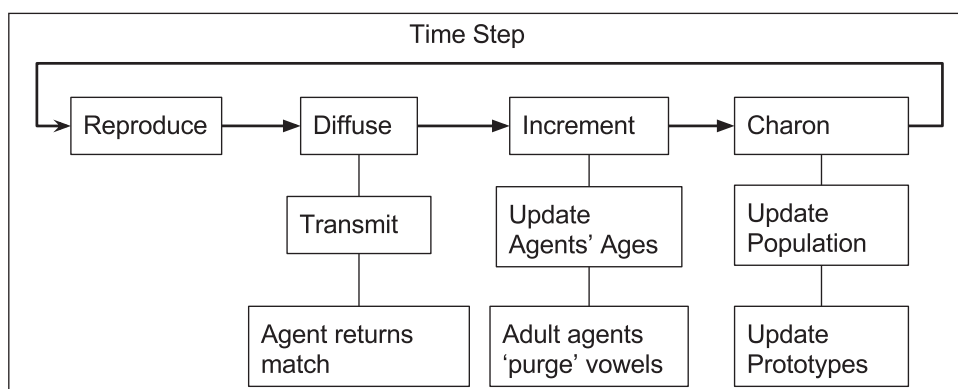


Figure 1: Action of a single time step.

2. Diffusion

Agents speak to each other in order to learn and pass on their language. In the current setup, only the vowel sounds in the various words are changeable during the learning process—the consonants are held fixed and are not represented as sound. For the first 10% of their lifespan (i.e. the number of steps an agent lives, which is a user parameter), agents hear the complete vocabulary of each family member, plus a random selection of words from the rest of the population. Learners add vowel cogphones to their repertoires and words to their vocabularies as they hear them.

3. Incrementation

All agents advance one step in age. Those *child* agents who reach the age of maturation discard the vowel cogphones in their repertoires that are not being used in words. After that point, mature agents speak to learners,

but no longer make changes to their vocabularies or vowel repertoires.

4. Charon

Agents who have reached the age limit are removed, and the *vowels convention* is calculated. This convention consists of population average pronunciations of the vowels in each of the various words.

Interface

VoSS runs in Python, and has a command-line interface with simple command keywords to change the parameters before running a simulation. Users can select a base *vowels convention* for the ancestors from a number of pre-sets modelling a variety of natural languages, or by entering a list of vowels using labels derived from the International Phonetic Alphabet (see **Figures 2** and **3**). The prototypical vowels are statically color-coded for the simulation.

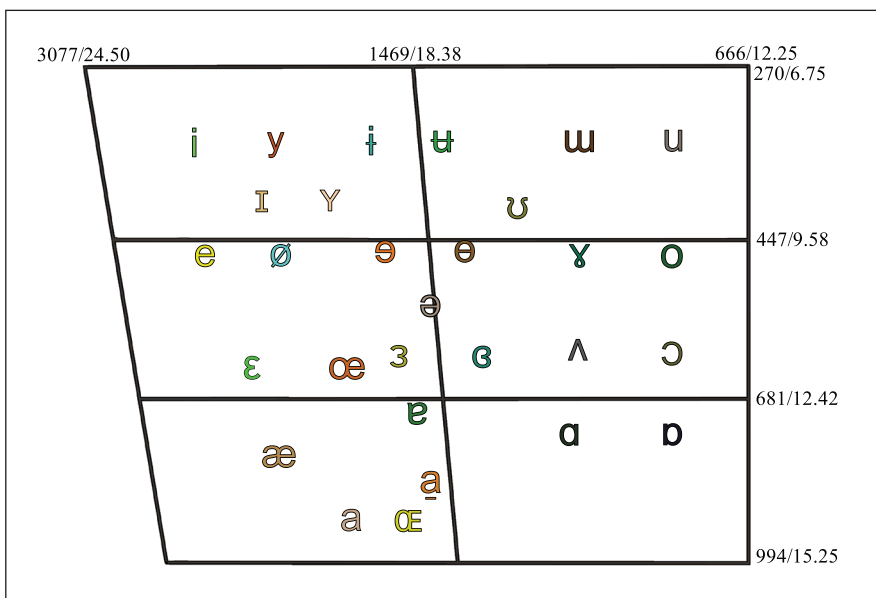


Figure 2: The full IPA chart. Axis labels show frequency in Hz/ERB.

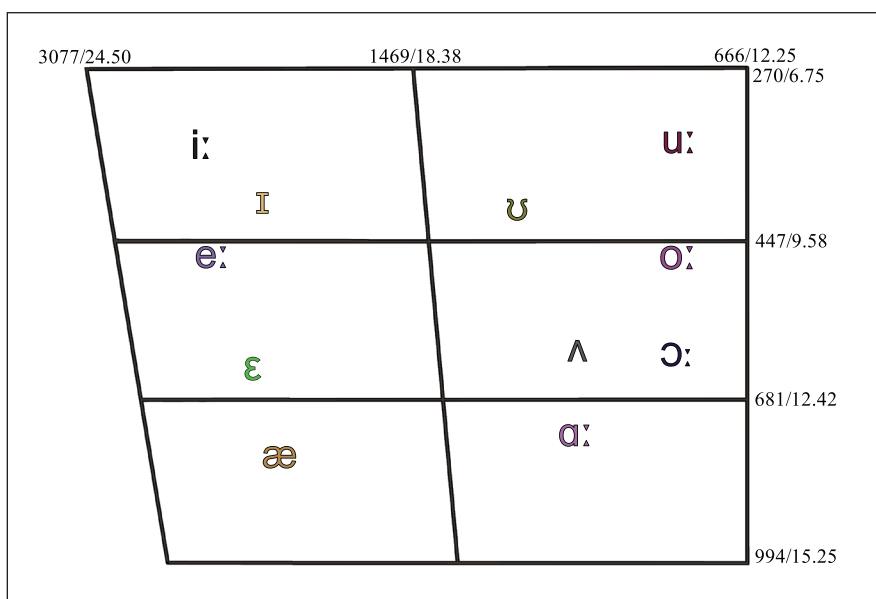


Figure 3: English vowel prototypes.

The language lexicon is generated randomly for each simulation, with each word consisting of a syllable onset, a vowel from the base convention, and a syllable coda. Each vowel is assigned to a random number of words, so that any vowel may have up to five times as many words in the lexicon as any other. This is an implementation of the concept of *functional load* in natural languages, in which different vowels occur in different numbers of words [16, 17].

Vowels are represented as 3-vectors comprising the first and second formant frequencies (F1 and F2) together with the length in milliseconds. The vowel formant space is represented in auditory frequency units of Equivalent Rectangular Bandwidth, which are converted to and from the more familiar frequency values in Hz using Traunmüller's [18] formula. This type of auditory formant space is based on studies of vowel perception and production [5].

A small number of consonants are hard-coded in the program, and are used at random to generate the words in the lexicon. Each consonant has a number of associated articulatory features which affect the vowels in production by *coarticulation*, and in perception by *deassimilation*. All words are monosyllabic, with one vowel and up to one consonant on either end, and there are no homophones at the beginning of the simulation (although homophones can form as it progresses).

At the end of each time step, the vowel chart is updated with the current live adults' pronunciation of the vowels in each of the words, which retain their color-coding throughout the simulation. Change in the language's vowel inventory can be observed as the average pronunciations move around the formant space. Users can opt to watch these results live with each time step, or turn off the graphics and get results after a set number of steps.

In each interaction (see **Figure 4**), agents imperfectly speak vowels from their internal repertoire in the context of mono-syllabic words. The listening agent may find a match in its repertoire or, if it finds none, will add a new phone within a similar latitude. The degrees of random imperfection, both for production of phones into spoken vowels and conceptualization by the listener, can be adjusted by the user.

Output

A realistic aspect of the simulation is that a vowel is not a singular kind of entity, but rather has distinct identities both as a physical sound and as a cognized "known" vowel.

The former entity is what we call a *vowel*, while the second entity is what we call a phone or *cogphone*. The simulation is able to show either of these entities. With phone sampling, the graphics output shows a small colour-coded dot for each phone in each agent's repertoire, and larger coloured circles for the averages of these phones (see **Figure 5**). The phones are affected by *deassimilation* (adjustment for the consonants) within each word context. The visualization reflects what the agents "know."

With vowel sampling, the graphic shows one small, coloured dot for each agent's pronunciation of each word at the time of sampling. Larger coloured circles show the average pronunciation of each word across all adult agents in the community. These reports reflect what the agents "say" and are affected by *assimilation* to the consonant context of each word.

For both options, the user can view a "shifting report" which shows the original prototype positions in black and the current average pronunciations in colour, so that the shifting distance and occurrence of lexical mergers/splits can be observed. These results can also be saved as a text file or as eps figures showing the resulting vowel space.

VoSS can also run extended simulations and collect output automatically. After setting the initial simulation parameters, the archiver prompts for a vowel system and then runs through 100 cycles. The program saves the phone and vowel shifting and sampling charts as eps images at every other step, and also writes a text file detailing the changes in average formant values over time.

Menu-accessible Parameters

The *perceptual margin* determines the maximum Euclidean distance in the formant space within which an agent will recognize an incoming signal as a match for a phone already existing in its repertoire. *Phone noise* defines a radius which acts as a margin of error in the formant space within which an agent internalizes knowledge of a *phonorm* (articulatory formation of a vowel in context). *Vowel noise* defines a radius within which an agent may speak a vowel example of an internal phonorm. These perceptual margin and noise parameters are an interpretation of Ohala's [6] "hyper-correction/hypo-correction" and Blevins' [19] Evolutionary Phonology models, wherein listeners compensating for speaker output variation are a driving force of change. It must be emphasized that these parameters are micro-level only, and so reflect only the agents' perception and production,

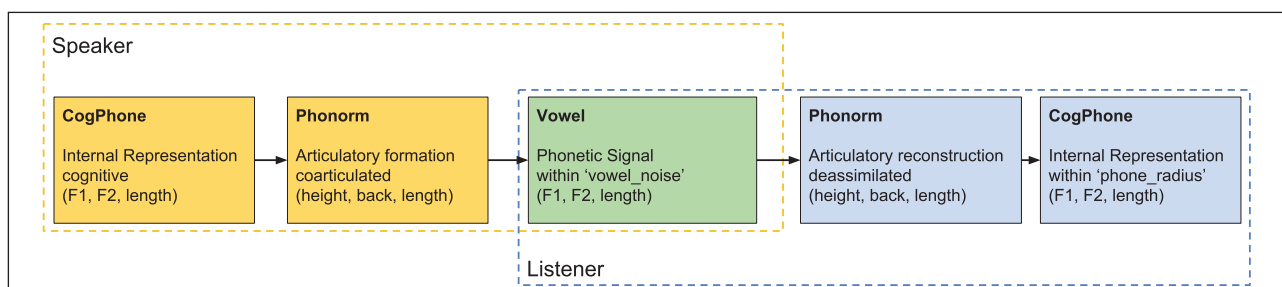


Figure 4: Schematic model of an interaction between speaker and listener. Formants F1 and F2 are represented in auditory scales.



Figure 5: Results of a 100-cycle simulation with Welsh inventory showing cogphone sampling.

not any meta-analysis of the linguistic system such as vowel contrast or system crowding etc. *Proximity add-on*, when positive, determines a radius in the formant space for triggering conflicts between phones in an agent's repertoire. The theoretical basis for the proximity margin dates back to 1952 [20].

Family size provides the number of agents who teach babies throughout their lifetime as learners. This parameter heavily affects the runtime of the simulation and also the scale and clustering of the language acquisition network. *Contacts* provides the number of randomly selected agents who speak to learners at each step. *Words per contact* limits the number of words a learner hears from each randomly selected contact. *Lifespan* determines the number of steps an agent will remain in the simulation.

The flag *show* set on will make the simulation update the graphical output with each step, which although interesting to follow, greatly increases the run time. *Phone/vowel* sampling allows the user to choose graphics output showing either the physical vowel sounds or the cogphones. *Color-coding* is optional and can be turned off. *Symbols* are optional; prototypes can be shown as filled circles or IPA symbols. *Micro viewing mode* highlights a single learning agent's repertoire in the graphic output and documents that agent's interactions, including all vowels it hears, and changes that occur in transmission.

Vocabulary reports are more detailed text output showing the agents' full repertoires and phone-word pairings. These can be used to track lexical diffusion of vowel shifting, mergers or splits. *Lexicon report* shows the vowels which agents currently have mapped to the original words in the lexicon. *Lexicon size* sets the minimum number of words in the language (which may be up to 5 times larger in its final form).

Learners/Teachers allows the user to switch between showing reports (text/graphical) for the learners only or for adults only (default is adults only). *Armchair agents* gives the user control over whether agents continue to manipulate their repertoires after the first step of their lifespan.

Quality control

VoSS is normally run from within a Python environment, and provides a command-line interface to first make desired changes to the default parameters, and to then start a simulation running (see use case in section 3 for more details). The appearance of the running simulation will vary depending on the *show* flag and whether vowels or phones are tracked. In any case the Python shell will state 'stepping' while the results of the next step are being calculated, and then the numerical version of the graphical output is printed, and the graphics updated if *show* is on. A typical simulation with reasonable parameters can easily take on the order of one to ten hours to complete on a typical consumer-level computer using Intel core i7 with multiple CPU (see **Figures 6** and **7**).

(2) Availability

Operating system

Linux (all modern distributions), Windows (7 and higher) and macOS.

Programming language

Python 3.4+

Additional system requirements

A system with at least 8GB of memory is recommended to run extended simulations. VoSS does not require any non-standard input or output devices.

Dependencies

Tkinter is required to be installed.
<https://docs.python.org/3/library/tkinter.html>

List of contributors

1. Hannah Scott (Developer).
2. Sean Fulop (Project Manager).

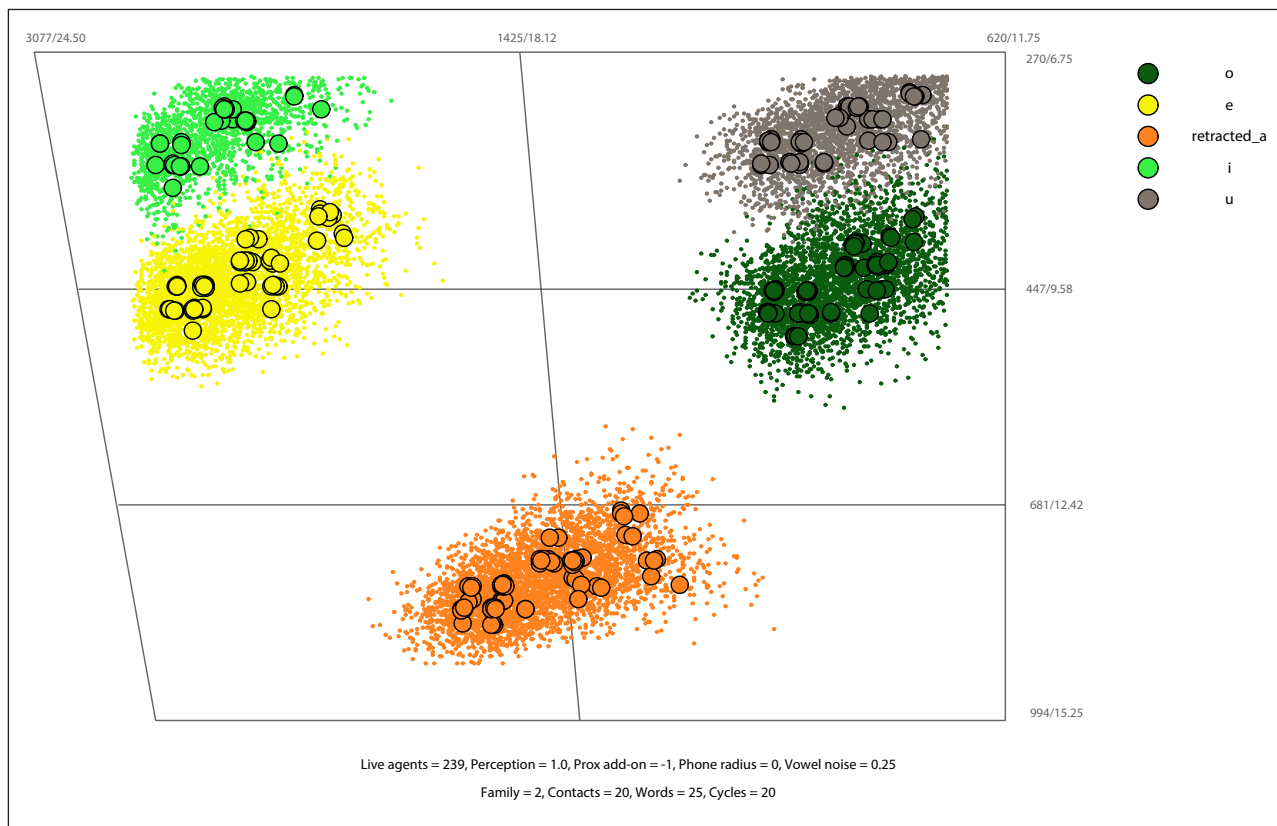


Figure 6: A simulation starting from Spanish vowels showing vowel sampling. A GIF animation showing the steps can be viewed here: https://raw.githubusercontent.com/Language-Science-Models-Lab/voss/master/spanish_vowels.gif. The file is also available for download here: <https://doi.org/10.5334/jors.198.s1>.

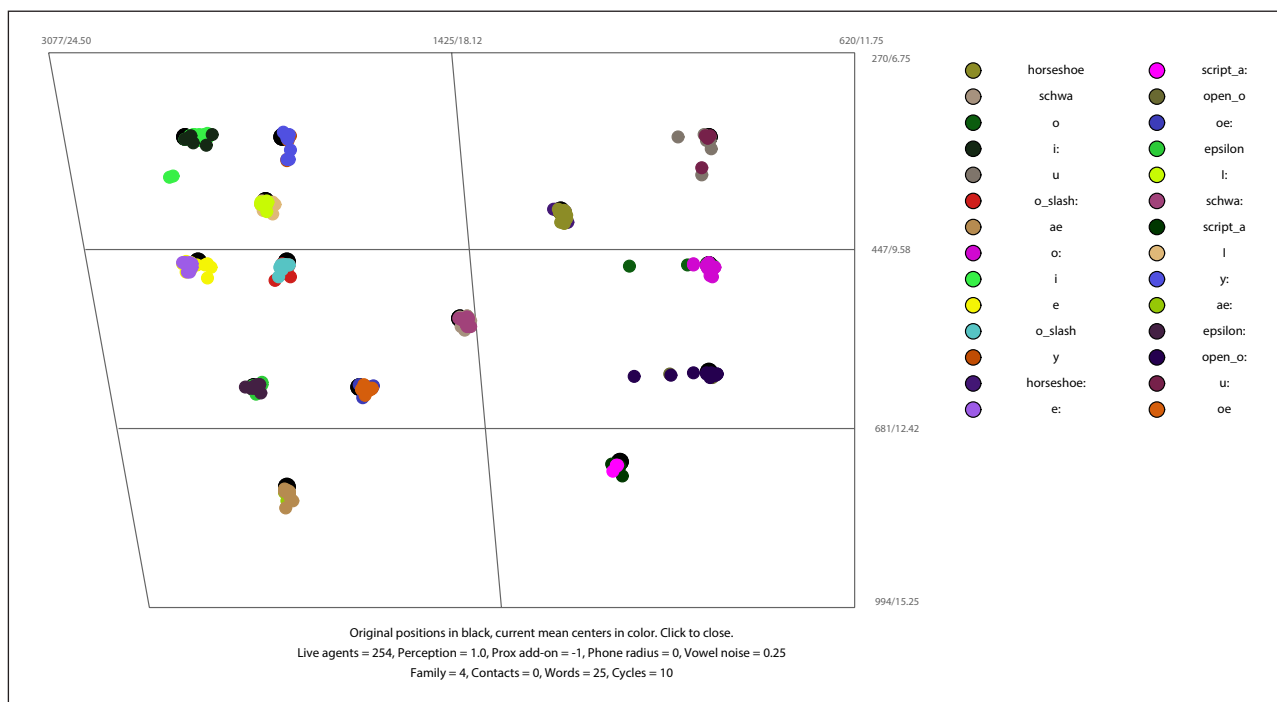


Figure 7: A simulation starting from Danish vowels showing cogphone shifting. A GIF animation showing the steps can be viewed here: https://raw.githubusercontent.com/Language-Science-Models-Lab/voss/master/danish_phones_shift.gif. The file is also available for download here: <https://doi.org/10.5334/jors.198.s2>.

Software location

Name: Vowel System Sandbox: A Complex Systems Model of Language Change

Persistent identifier: <https://zenodo.org/record/2532817>

Licence: GNU General Public License v3.0

Publisher: Hannah Scott

Version published: 3.9

Date published: 1/16/19

Code repository

Name: voss

Identifier: <https://github.com/Language-Science-Models-Lab/voss>

Licence: GNU General Public License v3.0

Date published: 1/16/19

Language

English

(3) Reuse potential

Linguistics as a field is generally stuck in the armchair when it comes to theorizing about language change. There is essentially no available software that could enable a legitimate computational science of language change to develop. VoSS is a first step in this direction, to allow linguists to test hypotheses and theories of the causes of sound change in language, and also to test and establish some of the basic parameters of the speech transmission process in humans.

The data which would be needed to analyse change at this level would be impractical to collect in real life, but modelling provides a good alternative to interact with the entire system. Moreover, researchers could potentially expand the program by adding mechanics from other theoretical viewpoints such as social implementation of changes, and language contact effects.

Typical use case

1. User indicates that she wants to run a simulation using the default parameters.
2. The VoSS software will show the parameters and display the dynamically-generated lexicon.
3. VoSS will present the base convention with summary of parameters below a vowel chart indicating the live vowel convention.
4. The user will confirm (via mouse-click in the chart) that they are ready to begin the simulation.
5. VoSS will print the lexicon with average vowel pronunciations among adult agents and present the convention plot with full adult sampling and averages at step-wise intervals until the requisite number of cycles is complete.
6. The user will confirm that they are ready to proceed.
7. The system will present the final convention juxtaposed with the base convention averages without individual sampling.
8. User confirms via mouse-click in the chart that she is finished viewing the final output.
9. The system will close the chart and maintain the current simulation.

Support will be offered as possible by the developers. The software will be updated on GitHub.

Additional Files

The additional files for this article can be found as follows:

- **Figure 6.** A simulation starting from Spanish vowels showing vowel sampling. A GIF animation showing the steps can be viewed here: https://raw.githubusercontent.com/Language-Science-Models-Lab/voss/master/spanish_vowels.gif. The file is also available for download here: <https://doi.org/10.5334/jors.198.s1>
- **Figure 7.** A simulation starting from Danish vowels showing cogphone shifting. A GIF animation showing the steps can be viewed here: https://raw.githubusercontent.com/Language-Science-Models-Lab/voss/master/danish_phones_shift.gif. The file is also available for download here: <https://doi.org/10.5334/jors.198.s2>

Acknowledgements

Thanks are owed to Chris Golston, Darin Flynn, Mark Ryan, Zach Metzler, and Brian Moran for discussions that furthered the ideas in the project.

Competing Interests

The authors have no competing interests to declare.

References

1. **Steels, L** 2000 Language as a complex adaptive system. In: *Parallel Problem Solving from Nature PPSN VI*, Schoenauer, M, et al. (eds.), 17–26. Springer. DOI: https://doi.org/10.1007/3-540-45356-3_2
2. **Ellis, N C** and **Larsen-Freeman, D** 2009 *Language as a Complex Adaptive System*. John Wiley & Sons.
3. **Kretzschmar Jr., W A** 2015 *Language and Complex Systems*. Cambridge University Press.
4. **Nicolis, G** and **Nicolis, C** 2012 *Foundations of Complex Systems*, 2nd ed. New Jersey: World Scientific. DOI: <https://doi.org/10.1142/8260>
5. **Rosner, B S** and **Pickering, J B** 1994 *Vowel Perception and Production*. Oxford University Press. DOI: <https://doi.org/10.1093/acprof:oso/9780198521389.001.0001>
6. **Ohala, J J** 1981 The listener as a source of sound change. In: *Chicago Linguistic Society*, Masek, C S, Hendrick, R A and Miller, M F (eds.), 178–203. University of Chicago Press.
7. **Ohala, J J** 1996 Speech perception is hearing sounds, not tongues. *The Journal of the Acoustical Society of America*, 99(3): 1718–1725. DOI: <https://doi.org/10.1121/1.414696>
8. **Hamann, S** 2015 Phonological Changes. In: *The Routledge Handbook of Historical Linguistics*, Bowerman, C and Evans, B (eds.), 249–263. Routledge: London.
9. **De Boer, B** 2001 *The Origins of Vowel Systems*, 1. Oxford University Press on Demand.
10. **Harrison, K D**, **Dras, M** and **Kapicioglu, B** 2002 Agent-based modeling of the evolution of vowel harmony. In: *North East Linguistic Society*.

11. **Winter, B** and **Wedel, A** 2016 The Co-evolution of Speech and the Lexicon: The Interaction of Functional Pressures, Redundancy, and Category Variation. *Topics in cognitive science*, 8(2): 503–513. DOI: <https://doi.org/10.1111/tops.12202>
12. **Chirkova, K** and **Gong, T** 2014 Simulating vowel chain shift in Xumi. *Lingua*, 152: 65–80. DOI: <https://doi.org/10.1016/j.lingua.2014.09.009>
13. **Harrington, J** and **Schiel, F** 2017 /u/-fronting and agent-based modeling: The relationship between the origin and spread of sound change. *Language*, 93(2): 414–445. DOI: <https://doi.org/10.1353/lan.2017.0019>
14. **Kirby, S**, **Griffiths, T** and **Smith, K** 2014 Iterated learning and the evolution of language. *Current Opinion in Neurobiology*, 28: 108–114. DOI: <https://doi.org/10.1016/j.conb.2014.07.014>
15. **Wilensky, U** 2017 *NetLogo*. Evanston, Illinois.
16. **Hockett, C F** 1967 The quantification of functional load. *Word*, 23(1–3): 300–320. DOI: <https://doi.org/10.1080/00437956.1967.11435484>
17. **Surendran, D** and **Niyogi, P** 2006 Quantifying the functional load of phonemic oppositions, distinctive features, and suprasegmentals. *Amsterdam Studies in the Theory and History of Linguistic Science*, 279. DOI: <https://doi.org/10.1075/cilt.279.05sur>
18. **Traunmüller, H** 1990 Analytical expressions for the tonotopic sensory scale. *The Journal of the Acoustical Society of America*, 88(1): 97–100. DOI: <https://doi.org/10.1121/1.399849>
19. **Blevins, J** 2004 *Evolutionary Phonology: The Emergence of Sound Patterns*. Cambridge University Press. DOI: <https://doi.org/10.1017/CBO9780511486357>
20. **Martinet, A** 1952 Function, structure, and sound change. *Word*, 8(1): 1–32. DOI: <https://doi.org/10.1080/00437956.1952.11659416>


How to cite this article: Fulop, S A and Scott, H 2019 Vowel System Sandbox: Complex System Modelling of Language Change. *Journal of Open Research Software*, 7: 8. DOI: <https://doi.org/10.5334/jors.198>

Submitted: 02 October 2017

Accepted: 18 February 2019

Published: 25 March 2019

Copyright: © 2019 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

 *Journal of Open Research Software* is a peer-reviewed open access journal published by Ubiquity Press.

OPEN ACCESS 