ACSIJ Advances in Computer Science: an International Journal, Vol. 3, Issue 1, No.7 , January 2014
ISSN : 2322-5157
www.ACSIJ.org

ACSIJ
WWW.ACSIJ.ORG

Digital Organism Simulation Environment (DOSE): A Library for Ecologically-Based *In Silico* Experimental Evolution

Clarence FG Castillo[1], and Maurice HT Ling[2]

[1] School of Information Technology, Republic Polytechnic
Singapore, Republic of Singapore
clarence.castillo_33@yahoo.com

[2] School of Chemical and Biomedical Engineering, Nanyang Technological University
Singapore, Republic of Singapore
Department of Zoology, The University of Melbourne
Parkville, Victoria 3010, Australia
mauriceling@acm.org

Abstract

Testing evolutionary hypothesis in biological setting is expensive and time consuming. Computer simulations of organisms (digital organisms) are commonly used proxies to study evolutionary processes. A number of digital organism simulators have been developed but are deficient in biological and ecological parallels. In this study, we present DOSE (Digital Organism Simulation Environment), a digital organism simulator with biological and ecological parallels. DOSE consists of a biological hierarchy of genetic sequences, organism, population, and ecosystem. A 3-character instruction set that does not take any operand is used as genetic code for digital organism, which the 3-nucleotide codon structure in naturally occurring DNA. The evolutionary driver is simulated by a genetic algorithm. We demonstrate the utility in examining the effects of migration on heterozygosity, also known as local genetic distance.

Keywords: *Digital Organisms, Simulation Environment, Ecology Simulation, Migration, Genetic Distance.*

1. Introduction

Nothing in Biology makes sense except in the light of Evolution -- Theodosius Dobzhansky [1]

Nothing in Medicine makes sense, except in the light of Evolution -- Ajit Varki [2]

Evolution is a fundamental aspect of biology. However, testing evolutionary hypotheses is a challenge [3] as it is highly time consuming and expensive, if not impossible. Long generation time associated with most species makes it virtually impossible to test evolutionary hypotheses in a laboratory setting. The longest on-going laboratory experiment in evolutionary biology have been initiated by Richard Lenski in 1988 [4], using a common intestinal bacterium, *Escherichia coli*, which has one of the shortest generation time. Other experimental evolution experiments [5-7], such as adaptation to salt and food additives, have also used *E. coli* due to its generation time. Despite so, it is generally prohibitively expensive to examine the genetic makeup of each bacterium using experimental techniques, such as DNA sequencing. At the same time, such examination is destructive in nature and the examined bacterium cannot be revived for further evolutionary experiments.

A means around these limitations is to use models of bacteria or higher organisms, rather than real biological organisms. These modeled organisms are known as artificial life or digital organisms (DO) which organisms are simulated, mutated, and reproduced in a computer [8]. Although digital organisms are not real biological organism, it has characteristics of being a real living organism but in a different substrate [9]. Batut et al. [3] argue that DO is a valuable tool to enable experimental evolution despite its drawbacks as repeated simulations can be carried out with recording of all events. Furthermore, only computational time is needed to study every organism, which is analogous to sequencing every organism, and this process is not destructive in a biological sense as the studied organism can be "revived" for further simulations.

The main tool needed for using DO is a computational platform to act as a simulation environment. A number of DO platforms have been developed [10]. One of the early simulators is Tierra [11], where each organism is an evolvable, mating and reproducing program competing for computing resources, such as CPU cycles and memory

space. Hence, Tierra's programs can be seen as an executable DNA. A major drawback of Tierra is that the DOs are not isolated from each other as all DOs shared and compete for the same memory space. Avida [12] simplified Tierra [11] by enabling each DO to run on its own virtual machine; thus, isolating each DO, resulting in CPU cycle being the main competing resource. As Tierra [11] and Avida [12] used bytecodes as the genetic constituents for DO, it is difficult to examine parameters such as heterozygosity and genetic distance, which is commonly used in population genetics [13] from HIV virus [14] to human migration [15]. Mukherjee et al. [16] defines heterozygosity as variation within population while genetic distance is the variation between populations. Hence, heterozygosity can be considered as local genetic distance or within group genetic distance. Aevol [3] used a binary string as genetic material and had incorporated concepts from molecular biology; such as genes, promoters, terminators, and various mutations; into its design. This allowed for genetic distance to be measured. However, aevol [3] is designed for simulating bacterial genetics and evolution. Hence, ecological concepts, such as migration and isolation, are not incorporated.

Previously, our group had designed a genetic algorithm (GA) framework conforming to biological hierarchy starting from gene to chromosome to genome (as organism) to population [17], which may help interpreting GA results to biological context. Further work [18, 19] by our group had formalized a 3-character genetic language to correspond the 3-nucleotide codon in naturally occurring DNA and incorporating a 3-dimensional "world" consisting of ecological cells in order to give it parallels to biological DNA and natural ecosystem.

Here, we present a Python DO simulation library, Digital Organism Simulation Environment (DOSE), built on our previous work [17-19]. We then illustrate the use of DOSE to examine the effects of migration on heterozygosity (local genetic distance) where DOs can only mate within their own ecological cell.

2. Methods

2.1 DOSE Library

The basis of DOSE is a simulation driver and management layer built on top of 4 different sets of components, which had been previously described [17-19]. The 4 sets of components are briefly described as follow; firstly, DOSE consists of a set of objects representing a chromosome, organism, and population [17]. An organism can consist of one or more chromosome to make up its genome and a population consists of one or more organisms. Secondly, a GA acts as the evolutionary driver acting on the chromosomes. Thirdly, Ragaraja interpreter [19] is used to read the chromosomes and update the cytoplasm (cell body). This resembles the translation of genes into proteins in biological context; hence, Ragaraja interpreter [19] can be seen as the transcription and translation machinery. Lastly, a 3-dimensional world [18] consisting of ecological cells allows the mapping of DOs onto the world.

Each simulation is defined by a set of parameters and functions, which are used by the simulation driver and management. It constructs and initializes the DOs, maps the DOs onto the world, runs the simulation from first generation to the maximum generation as defined in the parameter, and report the events into a text file or database as required. After DO initialization, the current simulation driver simulates each organism and ecological cell sequentially [18].

The following is the core set of 18 parameters available in DOSE to cater for various uses:

- *population_names*: provides the names of one or more populations
- *population_locations*: defines the deployment of population(s) at the start of the simulation
- *deployment_code*: defines the type of deployment scheme
- *chromosome_bases*: defines allowable bases for the genetic material
- *background_mutation*: defines background mutation rate
- *additional_mutation*: defines mutation rate on top of background mutation rate
- *mutation_type*: defines a default type of mutation
- *chromosome_size*: defines the initial size of each chromosome
- *genome_size*: defines the number of chromosome(s) in each organism
- *max_tape_length*: defines the size of cytoplasm
- *interpret_chromosome*: defines whether phenotype is to be simulated
- *max_codon*: defines the maximum number of codons to express
- *population_size*: defines the number of organisms per population

- *world_x, world_y, world_z*: defines the size of the world in terms of numbers of ecological cells
- *maximum_generations*: defines the number of generations to simulate
- *ragaraja_instructions*: list of recognized codons

The following is the core set of 12 functions definable in DOSE to cater for various uses; of which, Functions 2 to 11 were previously defined [18]:

1. *deployment_scheme*: initial deployment of organisms into the ecosystem
2. *fitness*: calculates the fitness of the organism and returns a fitness score
3. *mutation_scheme*: mutation events in each chromosome
4. *prepopulation_control*: population control events before mating event in each generation
5. *mating*: mate choice and mating events
6. *postpopulation_control*: population control events after mating event in each generation
7. *generation_events*: other irregular events in each generation
8. *organism_movement*: short distance movement of organisms within the world, such as foraging
9. *organism_location*: long distance movement of organisms within the world, such as flight
10. *ecoregulate*: events to the entire ecosystem
11. *update_ecology*: local environment affecting entire ecosystem
12. *update_local*: ecosystem affecting the local environment

2.2 Simulations

Two sets (Example 1 and Example 2) of three simulations with different migration schemes; no migration, adjacent migration, and long migration; were developed, giving a total of six simulations. Each simulation consisted of a 25-cell flat world with 50 organisms per cell and mating could only be carried out between organisms within the same cell. As a result, each cell resembled an isolated landmass. One binary chromosome of 5000 bases formed the genetic material for each organism. Only point mutation was used and the two sets of simulation differ by point mutation rates. In the first set of 3 simulations (Example 1), mutation rate was set at 0.001, resulting in 5 point mutations per generation. In the second set of simulations (Example 2), mutation rate was set at 0.002, effectively doubling the occurrence of point mutations per generation compared to Example 1. Since the chromosomes were binary, mutation events were limited to inverting the base from one to zero and vice versa. Mutation scheme was identical in all 3 migration schemes. In no migration simulation, organisms were not allowed to cross cell boundaries throughout the simulation in order to simulate complete isolation. In adjacent migration simulation, 10% of the organisms from a cell can migrate to one of its 8-neighbour cell within a generation in order to simulate short distance migration patterns, such as foraging or nomadic behavior. In long migration, 10% of the organisms from a cell can migrate to any other cells within a generation in order to simulate long distance migration patterns, such as flight. Each simulation was performed for 1000 generations.

2.3 Data Analysis

Within cell analyses were performed. Hamming distance [20] was calculated between the chromosomes of two organisms and used as local genetic distance. 50 random pairs of organisms within a cell were selected for pair-wise local genetic distance calculation and an average heterozygosity was calculated for each cell in every generation. Within a generation, mean and standard error of heterozygosity were calculated from the average local genetic distances of 25 cells for each simulation.

3. Results

In this study, we present Python DO simulation library, Digital Organism Simulation Environment (DOSE), built on our previous work [17-19]. We first briefly outline a typical use of a DO simulation platform such as DOSE before illustrating 2 examples to examine the effects of migration on heterozygosity, given that the DOs can only mate within their own ecological cell.

3.1 Typical use of an *in silico* evolutionary platform

Similar to other *in silico* evolutionary platforms such as aevol [3], the basic output of DOSE is a set of time series data with generation count as the timing variable. These can include organism statistics; such as fitness, and genome size; or population statistics; such as average fitness, and genetic distance. Further analyses can be carried out from these results. For example, if the parent-child (also known as ancestry) relationships are recorded, the lineage of beneficial mutations can be carried out using genealogical analysis [21]. Further studies using 2 or more evolved populations of digital organisms, such as measuring mutational robustness using a competition [22, 23], may be performed. These competition assays may be used to model biological processes, such as parasitism [24].

A typical *in silico* evolutionary experiment consists of modifying one or more parameters, such as mutation rate,

and/or functions, such as mating scheme, in the platform, and examining the time series data emerging from one or more simulations. Batut et al. [3] highlighted that fortuitous events can be distinguished from systematic trends by comparing data from replicated simulations. It is also possible to revive one or more simulations from stored data and that can be mixed to simulate interactions between groups of organisms [25].

3.2 Example 1: Testing the effects of migration on heterozygosity

DOSE is designed as a tool to examine evolutionary scenarios on an ecological setting. In this example, we examine the effects of migration, simulated by movement of organisms to adjacent or across non-adjacent ecological cells.

Hamming distance [20], which had been used as distance measure for phylogenetic determination between viruses [26, 27], was used in this study as a measure of heterozygosity. As chromosomal lengths were identical in all organisms throughout the simulation, Hamming distance represented the number of base differences between any two organisms.

Our results show that the average heterozygosity for no migration and long migration across all 1000 generations for all 25 ecological cells is similar (p-value = 0.989; Table 1). The average heterozygosity for adjacent migration is marginally lower but not significantly different from that of no migration (p-value = 0.932) or long migration (p-value = 0.921). The average spread (standard error) of heterozygosity for no migration and long migration is also similar (p-value = 0.264; Figure 1A and 1C). However, the spread of heterozygosity for adjacent migration is significantly larger (p-value < 4.3×10^{-26}), especially after 500 generations (Figure 1B).

Table 1: Summary statistics of 3 simulations with mutation rate of 0.001

Simulation	*Average Heterozygosity*	*Average Standard Error*
No migration	1228.19	25.064
Adjacent migration	1226.09	32.664
Long migration	1228.54	25.661

The average spread of heterozygosity from organisms within an ecological cell can be used as a proxy to estimate the variation within local population or intra-population [28]. Our results suggest that adjacent migration between sub-groups of mating populations results in the increase of genetic variation within local populations. The scenario of no migration acts as a control and long migration scenario yields the same local population variation as control where genetic variation only occurs from mutations. This suggests that long distance migration covering the entire ecosystem may result in the entire ecosystem behaving as one geographically extensive "local" population. This is observed in hoverflies where extensive migration result in the lack of genetic differentiation in a continental scale [29]. A study in human populations also suggested that long migration may result in the lack of genetic variation between sub-populations [30], which is consistent with our simulation results.

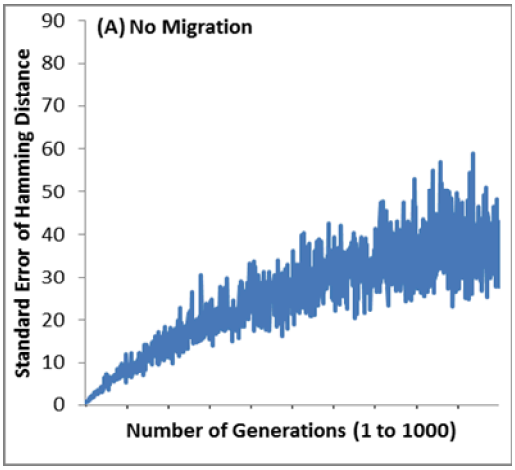

Fig. 1a Standard error of heterozygosity for no migration scenario.

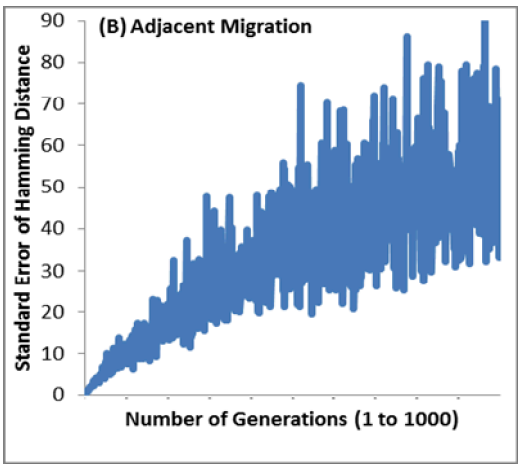

Fig. 1b Standard error of heterozygosity for adjacent migration scenario.

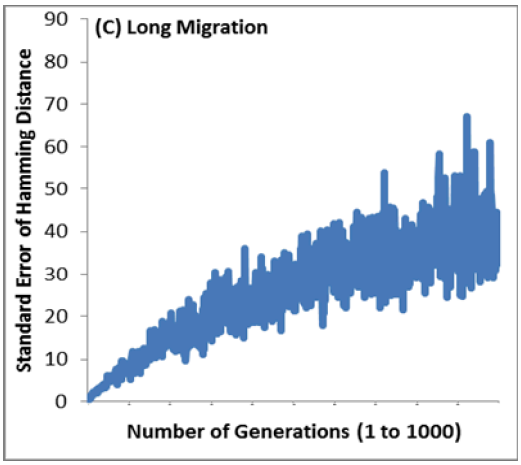

Fig. 1c Standard error of heterozygosity for long migration scenario.

Our results also suggest that migration and mating between adjacent sub-populations increased the genetic variability, as seen in increased variation between adjacent migration and no migration scenarios. This is supported by current study suggesting that migration is crucial in maintaining genetic variation [31].

3.3 Example 2: Testing the effects of mutation rates and migration on heterozygosity

In this example, we double the mutation rate from 0.001 (0.1%) to 0.002 (0.2%) on the 3 migration scenarios in Example 1. The simulation results can be analyzed in the same manner as Example 1 or compared with that of Example 1 to examine the effect of increased mutation rate.

Our results show that there is no difference in the average heterozygosity between all 3 simulations ($F = 0.01$, p-value = 0.987; Table 2). The spread of heterozygosity is significantly higher in adjacent migration when compared to no migration (p-value = 4.4×10^{-34}) or long migration (p-value = 2.2×10^{-31}) scenarios (Figure 2). These results are consistent with that of Example 1, suggesting that these trends are not significantly impacted by doubling the mutation rate.

Table 2: Summary statistics of 3 simulations with mutation rate of 0.002

Simulation	*Average Heterozygosity*	*Average Standard Error*
No migration	1787.79	36.296
Adjacent migration	1784.32	45.776
Long migration	1788.52	36.695

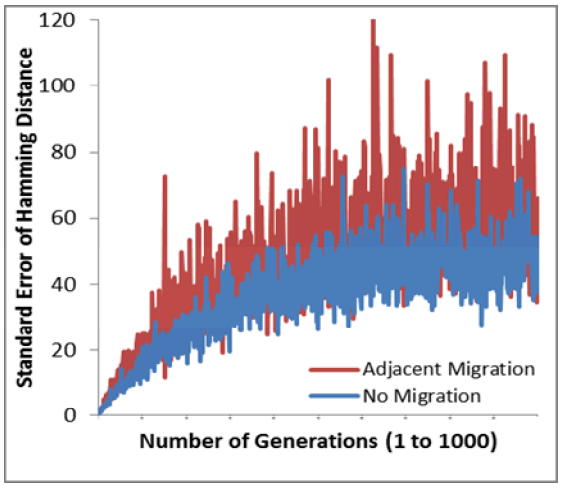

Fig. 2 Standard errors of heterozygosity between no migration and long distance migration for mutation rate of 0.002.

By comparing simulation outputs from different mutation rates (0.1% against 0.2%), our results show that heterozygosity (Figure 3A) and spread of heterozygosity (Figure 3B) are increased with higher mutation rate. This increase is significant for both heterozygosity (p-value < 6.8×10^{-90}) and spread of heterozygosity (p-value < 7.3×10^{-55}). However, the trend is consistent in both examples. This is consistent with Mukherjee et al. [16] whom demonstrates that mutation rates does not impact on the statistical tests for evaluating heterozygosity and genetic distance using a simulation study.

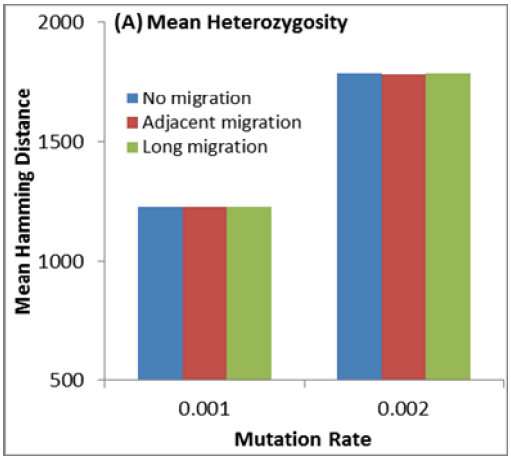

Fig. 3a Mean heterozygosity between migration scenarios for both mutation rates.

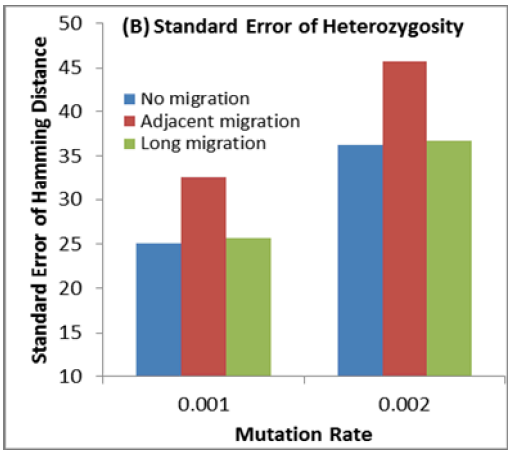

Fig. 3b Standard error of heterozygosity between migration scenarios for both mutation rates.

4. Conclusions

In this study, we have presented a Python DO simulation library, Digital Organism Simulation Environment (DOSE), built on our previous work [17-19]. DOSE is designed with biological and ecological parallels in mind. As a result, it is relatively easy to construct evolutionary simulations to examine evolutionary scenarios, especially when a complex interaction of environment and biology is required. To illustrate the use of DOSE in an ecological context, we have presented 2 examples on the effects of migration schemes on heterozygosity. Our simulation results show that adjacent migration, such as foraging or nomadic behavior, increases heterozygosity while long distance migration, such as flight covering the entire ecosystem, does not increase heterozygosity. These results are consistent with previous studies [29, 30].

Appendix

DOSE version 1.0.0 is released under GNU General Public License version 3 at http://github.com/mauriceling/dose/ release/tag/v1.0.0 and anyone is encouraged to fork from this repository. Documentation can be found at http:// maurice.vodien.com/project-dose.

Acknowledgement

The authors will like to thank AJZ Tan (Biochemistry, The University of Melbourne) and MQA Li (Institute of Infocomm Research, Singapore) for their comments on an initial draft of this manuscript.

References

[1] T. Dobzhansky, “Nothing in biology makes sense except in the light of evolution”, The American Biology Teacher, Vol. 35, 1973, pp. 125-129.
[2] A. Varki, “Nothing in medicine makes sense, except in the light of evolution”, Journal of Molecular Medicine, Vol. 90, 2012, pp. 481-494.
[3] B. Batut, D. P. Parsons, S. Fischer, G. Beslon and C. Knibbe, “*In silico* experimental evolution: a tool to test evolutionary scenarios”, BMC Bioinformatics, Vol. 14, No. Suppl 15, 2013, Article S11.
[4] J. E. Barrick and R. E. Lenski, “Genome dynamics during experimental evolution”, Nature Reviews Genetics, Vol. 14, No. 12, 2013, pp. 827-839.
[5] C. H. Lee, J. S. H. Oon, K. C. Lee and M. H. T. Ling, “*Escherichia coli* ATCC 8739 adapts to the presence of sodium chloride, monosodium glutamate, and benzoic acid after extended culture”, ISRN Microbiology, Vol. 2012, 2012, Article ID 965356.
[6] J. A. How, J. Z. R. Lim, D. J. W. Goh, W. C. Ng, J. S. H. Oon, K. C. Lee, C. H. Lee and M. H. T. Ling, “Adaptation of *Escherichia coli* ATCC 8739 to 11% NaCl”, Dataset Papers in Biology 2013, 2013, Article ID 219095.
[7] D. J. W. Goh, J. A. How, J. Z. R. Lim, W. C. Ng, J. S. H. Oon, K. C. Lee, C. H. Lee and M. H. T. Ling, “Gradual and step-wise halophilization enables *Escherichia coli* ATCC 8739 to adapt to 11% NaCl”, Electronic Physician, Vol. 4, No. 3, 2012, pp. 527-535.
[8] S. F. Elena and R. Sanjuán, “The effect of genetic robustness on evolvability in digital organisms”, BMC Evolutionary Biology, Vol. 8, 2008, pp. 284.
[9] Y. Z. Koh, and M. H. T. Ling, MHT, “On the liveliness of artificial life”, Human-Level Intelligence, Vol. 3, 2013, Article 1.
[10] A. Adamatzk, and M. Komosinski M, Artificial life models in software, London: Springer-Verlag, 2005.
[11] T. S. Ray, 1991, "Evolution and optimization of digital organisms", in Billingsley K.R. et al. (eds), Scientific Excellence in Supercomputing: The IBM 1990 Contest Prize Papers, 1991, pp. 489–531.
[12] C. Ofria, and C. O. Wilke, “Avida: A software platform for research in computational evolutionary biology”, Artificial Life, Vol. 10, 2004, pp. 191-229.
[13] O. Tal, “Two complementary perspectives on inter-individual genetic distance”, Biosystems, Vol. 111, No. 1, 2013, pp. 18-36.
[14] M. Arenas, and D. Posada, “Computational design of centralized HIV-1 genes”, Current HIV Research, Vol. 8, No. 8, 2010, pp. 613-621.
[15] S. J. Park, J. O. Yang, S. C. Kim, and J. Kwon, “Inference of kinship coefficients from Korean SNP genotyping data”, BMB Reports, Vol. 46, No. 6, 2013, pp. 305-309.
[16] M. Mukherjee, D. O. Skibinski, and R. D. Ward, “A simulation study of the neutral evolution of heterozygosity

and genetic distance", Heredity, Vol. 53, No. 3, 1987, pp. 413-423.
[17] J. Z. R. Lim, Z. Q. Aw, D. J. W. Goh, J. A. How, S. X. Z. Low, B. Z. L. Loo, and M. H. T Ling, "A genetic algorithm framework grounded in biology", The Python Papers Source Codes, Vol. 2, 2010, Article 6.
[18] M. H. T Ling, "An artificial life simulation library based on genetic algorithm, 3-character genetic code and biological hierarchy", The Python Papers, Vol. 7, 2012, Article 5.
[19] M. H. T Ling, "Ragaraja 1.0: The genome interpreter of Digital Organism Simulation Environment (DOSE)", The Python Papers Source Codes, Vol. 4, 2012, Article 2.
[20] R. W. Hamming, RW, "Error detecting and error correcting codes", Bell System Technical Journal, Vol. 29, No. 2, 1950, pp. 147–160.
[21] T. D. Cuypers, and P. Hogeweg, "Virtual genomes in flux: an interplay of neutrality and adaptability explains genome expansion and streamlining", Genome Biology and Evolution, Vol. 4, No. 3, 2012, pp. 212-229.
[22] R. E. Lenski, C. Ofria, T. C. Collier, and C. Adami, "Genome complexity, robustness and genetic interactions in digital organisms", Nature, Vol. 400, No. 6745, 1999, pp. 661-664.
[23] J. Sardanyés, S. F. Elena, and R. V. Solé, "Simple quasispecies models for the survival-of-the-flattest effect: The role of space", Journal of Theoretical Biology, Vol. 250, No. 3, 2008, pp. 560-568.
[24] F. M. Codoñer, J. A. Darós, R. V. Solé, and S. F. Elena, "The fittest versus the flattest: experimental confirmation of the quasispecies effect with subviral pathogens", PLoS Pathogens, Vol. 2, No. 12, 2006, Article e136.
[25] M. A. Fortuna, L. Zaman, A. P. Wagner, and C. Ofria C, "Evolving digital ecological networks", PLoS Computational Biology, Vol. 9, No. 3, 2013, Article e1002928.
[26] C. D. Pilcher, J. K. Wong, and S. K. Pillai, SK, "Inferring HIV transmission dynamics from phylogenetic sequence relationships", PLoS Medicine, Vol. 5, No. 3, 2008, Article e69.
[27] A. M. Tsibris, U. Pal, A. L. Schure, R. S. Veazey, K. J. Kunstman, T. J. Henrich, P. J. Klasse, S. M. Wolinsky, D. R. Kuritzkes, and J. P. Moore, "SHIV-162P3 infection of rhesus macaques given maraviroc gel vaginally does not involve resistant viruses", PLoS One, Vol. 6, No. 12, 2011, Article e28047.
[28] D. W. Drury, and M. J. Wade, "Genetic variation and co-variation for fitness between intra-population and inter-population backgrounds in the red flour beetle, *Tribolium castaneum*", Journal of Evolutionary Biology, Vol. 24, No. 1, 2011, pp. 168-176.
[29] L. Raymond, M. Plantegenest, and A. Vialatte, "Migration and dispersal may drive to high genetic variation and significant genetic mixing: the case of two agriculturally important, continental hoverflies (*Episyrphus balteatus* and *Sphaerophoria scripta*)", Molecular Ecology, Vol. 22, No. 21, 2013, pp. 5329-5339.
[30] J. H. Relethford, "Heterogeneity of long-distance migration in studies of genetic structure", Annals of Human Biology, Vol. 15, No. 1, 1988, pp. 55-63.
[31] M. Yamamichi, and H. Innan, "Estimating the migration rate from genetic variation data", Heredity, Vol. 108, No. 4, 2012, pp. 362-363.