HgtSIM: A simulator for horizontal gene transfer (HGT) in microbial communities

Weizhi Song [1, 2] , **Kerrin Steensen** [1, 3] , **Torsten Thomas** Corresp. [1, 4]

[1] Centre for Marine Bio-Innovation, University of New South Wales, Sydney, Australia
[2] School of Biotechnology and Biomolecular Sciences, University of New South Wales, Sydney, Australia
[3] Department of Genomic and Applied Microbiology, Georg-August Universität Göttingen, Göttingen, Germany
[4] School of Biological, Earth and Environmental Sciences, University of New South Wales, Sydney, Australia

Corresponding Author: Torsten Thomas
Email address: t.thomas@unsw.edu.au

The development and application of metagenomic approaches have provided an opportunity to study and define horizontal gene transfer (HGT) on the level of microbial communities. However, no current metagenomic data simulation tools offer the option to introduce defined HGT within a microbial community. Here, we present HgtSIM, a pipeline to simulate HGT event among microbial community members with user-defined mutation levels. It was developed for testing and benchmarking pipelines for recovering HGTs from complex microbial datasets. HgtSIM is implemented in Python3 and is freely available at: https://github.com/songweizhi/HgtSIM .

HgtSIM: A simulator for horizontal gene transfer (HGT) in microbial communities

Weizhi Song[1,2], Kerrin Steensen[1,3] and Torsten Thomas[1,4]

[1]Centre for Marine Bio-Innovation, [2]School of Biotechnology and Biomolecular Sciences and [4]School of Biological, Earth and Environmental Sciences, University of New South Wales, NSW 2052, Australia. [3]Department of Genomic and Applied Microbiology, Georg-August-University Göttingen, Grisebachstr. 8, D-37077 Göttingen, Germany.

Corresponding Author: Torsten Thomas (t.thomas@unsw.edu.au)

Abstract

The development and application of metagenomic approaches have provided an opportunity to study and define horizontal gene transfer (HGT) on the level of microbial communities. However, no current metagenomic data simulation tools offer the option to introduce defined HGT within a microbial community. Here, we present HgtSIM, a pipeline to simulate HGT event among microbial community members with user-defined mutation levels. It was developed for testing and benchmarking pipelines for recovering HGTs from complex microbial datasets. HgtSIM is implemented in Python3 and is freely available at: https://github.com/songweizhi/HgtSIM.

1 Introduction

Horizontal gene transfer (HGT) has been recognized as an important force in microbial evolution and adaptation (Soucy *et al.*, 2015). A number of pipelines have been developed to identify HGTs between draft or completed genomes of isolated microorganisms (Hasan *et al.*, 2012; Podell and Gaasterland, 2007; Zhu *et al.*, 2014). In recent years, the development and application of metagenomic approaches have provided novel and vast amounts of information on the genomic composition of uncultured microorganisms (Thomas *et al.*, 2012). This offers an opportunity to study HGT on the level of microbial communities, however new bioinformatics tools have to be developed to reliable detect any HGT events. Simulations of metagenomics reads have been essential for the development and benchmarking of pipelines for the quality control, assembly, binning and annotation of metagenomic data (Peng *et al.* 2012; Kang *et al.* 2015). These simulation tools typically produce reads based on defined sets of reference genomes with user-defined abundance distributions and often considering realistic errors models for common sequencing technologies (Escalona *et al.*, 2016). However, no current simulation tool offers the option to introduce defined HGT within the microbial community simulated, thus allowing to test pipelines that aim to detect HGT. Here, we have developed a pipeline called HgtSIM, which can simulate HGTs between the genomes of microbial community. The pipeline can simulate HGTs with different degrees of similarity for transferred genes found in donor and recipient genomes, thus allowing to assess the detection of relatively recent or past transfers.

2 Methods

2.1 Simulation of gene mutations

The transfer of genes into a recipient genome often involves subsequent mutations that reflect evolutionary drift or adaptation to the new cellular context (e.g. change in codon usage to match tRNA availability). To simulate such mutations without disrupting reading frames and to confine the mutations to a defined range, we use codons as units of mutations. The mutations of codons were grouped into four categories (C_i): 1) one-base, silent mutation; 2) one-base, non-silent mutation; 3) two-bases mutations and 4) three-bases mutations (**Table 1**).

Table 1 Mutation types of codons

Mutation type	Example
one-base, same aa mutations (C_1)	AT**C** (Ile) → AT**A** (Ile)
one-base, different aa mutations (C_2)	**G**CC (Ala) → **A**CC (Thr)
Two bases mutation (C_3)	C**TC** (Leu) → C**CT** (Pro)
Three bases mutation (C_4)	**GTG** (Val) → **TAC** (Tyr)

The changed bases are displayed in bold. The corresponding amino acid change is given in parenthesis.

The algorithm for simulating random mutations is as follows:

(1) Get the length (L) of each gene to be transferred.

(2) Define the number of bases need to be changed (N) based on a user-defined identity value (I) and L: i.e. N = LI/100.

(3) Define the type of mutations based on N and a user-defined ratio of the four mutation categories. For example, if a ratio of 1:1:1:1 is specified for C_1:C_2:C_3:C_4, then, $N = C_1 + C_2 + 2C_3 + 3C_4$.

(4) Randomly select C_1, C_2, C_3 and C_4 codons and perform the corresponding mutations.

All changed nucleotides are recorded in a mutation report file. A BlastP-based comparison between the amino acid sequences is also provided.

2.2 Simulation of gene transfers

The steps to simulate random gene transfers are as follows (**Fig. 1A**):

(1) Add flanking sequences (if specified) to the (mutated) genes to be transferred. These flanking regions could, for example, be transposon insertion sequences.

(2) Get the total length of the recipient genome (P) and user-defined number of genes (Q) to be transferred.

(3) Randomly select Q numbers between 1 and P and cut the recipient genome at these positions to create sub-sequences.

(4) Randomly assign the (mutated) gene to be transferred to the cut point and concatenate them with the sub-sequences.

All the break positions and the (mutated) genes inserted to these positions are recorded in an insertion report file.

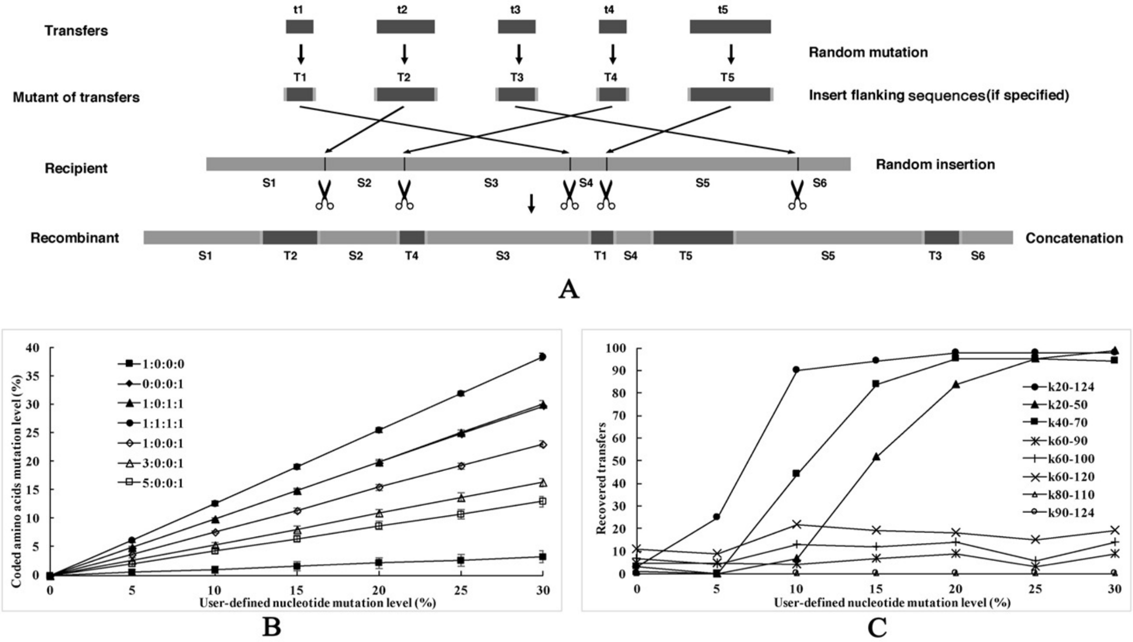

Fig. 1. (A) The workflow of HgtSIM. (B) The correlation of mutation on the nucleotide level and the resulting aa changes under different mutation category ratios. The four numbers separated by colon refer to the ratio between C_1, C_2, C_3 and C_4. (C) The effect of assembly k-mer sizes on the recovery of HGT events.

3 Results and Discussion

3.1 The effect of mutation categories on the level of aa changes

The correlation of mutation on the nucleotide level and the resulting aa changes under different ratios of mutation categories were assessed by performing random mutations on 100 genes selected from ten *Alphaproteobacteria* genomes (**Table 2**). The category ratios of “0:0:0:1” and “1:0:1:1” resulted in level of amino acid sequence changes that were similar to the user-defined level of nucleotide mutations (**Fig. 1B**). This correlation analysis provides the user information on the level of changes that occur at any given mutation level and category setting.

Table 2 The selected 20 genomes

Class	Strain	*NCBI BioProject ID*
Alphaproteobacteria	*Acidiphilium multivorum* AIU301	60101
	Ketogulonigenium vulgarum WSH 001	161161
	Mesorhizobium australicum WSM2073	47287
	Methylocapsa acidiphila B2	72841
	Methyloferula stellata AR4	165575
	Rhodovibrio salinarum DSM 9154	84315
	Roseobacter litoralis Och 149	19357
	Sphingobium japonicum UT26S 1	19949
	Starkeya novella DSM 506	37659
	Tistrella mobilis KA081020 065	76349
Betaproteobacteria	*Alicycliphilus denitrificans* K601	50751
	Dechlorosoma suillum PS	37693
	Gallionella capsiferriformans ES 2	32827
	Herbaspirillum seropedicae SmR1	47945
	Nitrosospira multiformis ATCC 25196	13912
	Ramlibacter tataouinensis TTB310	16294
	Sideroxydans lithotrophicus ES 1	33161
	Snodgrassella alvi wkB2	167602
	Sulfuricella denitrificans skB26	170011
	Tetrathiobacter kashmirensis WT001	67337

3.2 The effect of assembly k-mer range on the recovery of simulated HGTs

We next demonstrated the usefulness of HgtSIM to assess the recovery rate of HGTs within a community during a sequence assembly process. For this, 10 genes each from the 10 *Alphaproteobacteria* genomes were selected and randomly transferred to the 10 *Betaproteobacteria* genomes (**Table 2**) with various degrees of mutation (0%, 5%, 10%, 15%, 20%, 25% and 30%). GemSIM (McElroy *et al.* 2012) was used to simulate 10 million paired-ended 100-bp Illumina reads with 250 bp insert size from the 20 genomes for each mutation group. After quality filtering using Trimmomatic (Bolger *et al.* 2014) with a quality cutoff of 30 and a sliding window of 6 bp, the paired-ended reads were then assembled with IDBA_UD (Peng *et al.* 2012), a popular metagenome assembler, with multiple k-mer ranges. A gene transfer was considered to be recovered in the assembly if at least one of its two flanking regions is > 1 Kbp.

The results show that at low levels of mutations (≤5%), only a small proportion of transfers can be recovered. The "mink" value of IDBA_UD had a substantial impact on HGT detection. With values above 40 bp very low levels of recovery were shown, even when high numbers of mutations were introduced. The best recovery with more than 90% success for mutation levels of ≥10% was obtained with a full k-mer range (from 20 bp to 124 bp) (**Fig. 1C**) and this setting would thus be recommended to reconstruct regions of HGT in real metagenomic assemblies.

3.3 The effect of reads length and insert size on the recovery of HGTs with no/low mutation(s)

We also simulated how insert sizes and read length might influence recovery of transfer events. As at ≤5% mutation levels, only a small proportion (≤22%) of transfers were recovered by IDBA_UD (**Fig. 1C**). We focused on those mutation levels and simulated dataset with different read length (100 bp and 250 bp) and insert sizes (250 bp, 500 bp and 1 Kbp). The results show that at 0% mutation level, no improvement in recovery was observed with increased read length or insert sizes. For 5% mutation level, larger insert sizes improved recovery with 100 bp read length, but with 250 bp read length this was not observed (**Table 3**).

Table 3 The effect of reads length and insert size on the recovery of HGTs

Reads length (bp)	100			250		
Insert size (bp)	250	500	1000	250	500	1000
0%	2	2	0	0	0	0
5%	24	32	35	35	35	35

4 Conclusions

These examples demonstrate how various aspects of metagenomic sequencing projects (e.g. library production, read length, assembly parameters) can influence the potential to recover HGT from metagenomic datasets. Testing and benchmarking of various parameters and tools with simulated datasets produced by HgtSIM will in the future help to develop robust pipelines that have maximal success in recovering HGT from complex metagenomic data.

Acknowledgements

This research was funded by the Australian Research Council. Weizhi Song was funded by the China Scholarship Council (201508200019).

References

Bolger, A.M. *et al*. (2014) Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*, btu170.

Escalona, M. *et al*. (2016) A comparison of tools for the simulation of genomic next-generation sequencing data. *Nat. Rev. Genet.*, 17(8), 459-469.

Hasan, M.S. *et al*. (2012) GIST: Genomic island suite of tools for predicting genomic islands in genomic sequences. *Bioinformation*, 8(4), 203-205.

Kang, D.D. *et al*. (2015) MetaBAT, an efficient tool for accurately reconstructing single genomes from complex microbial communities. *PeerJ*, 3, e1165.

McElroy, K.E. *et al*. (2012) GemSIM: general, error-model based simulator of next-generation sequencing data. *BMC genomics*, 13(1), 74.

Peng, Y. *et al*. (2012) IDBA-UD: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. *Bioinformatics*, 28, 1420-1428.

Podell, S. and Gaasterland, T. (2007) DarkHorse: a method for genome-wide prediction of horizontal gene transfer. *Genome biology*, 8(2), R16.

Soucy, S.M. *et al*. (2015) Horizontal gene transfer: building the web of life. *Nature Rev. Genet.*, 16(8), 472-482.

Thomas, T. *et al.* (2012) Metagenomics-a guide from sampling to data analysis. *Microb. Inform. Exp.*, 2(1), 3.

Zhu, Q. *et al*. (2014) HGTector: an automated method facilitating genome-wide discovery of putative horizontal gene transfers. *BMC genomics*, 15(1), 717.