LEMON
a method to construct the local strains at horizontal gene transfer sites in gut metagenomics
Li, Chen; Jiang, Yiqi; Li, Shuaicheng

Published in:
BMC Bioinformatics

Published: 01/01/2019

Document Version:
Final Published version, also known as Publisher's PDF, Publisher's Final version or Version of Record

License:
CC BY

Publication record in CityU Scholars:
Go to record

Published version (DOI):
10.1186/s12859-019-3301-8

Publication details:

Citing this paper
Please note that where the full-text provided on CityU Scholars is the Post-print version (also known as Accepted Author Manuscript, Peer-reviewed or Author Final version), it may differ from the Final Published version. When citing, ensure that you check and use the publisher's definitive version for pagination and other details.

General rights
Copyright for the publications made accessible via the CityU Scholars portal is retained by the author(s) and/or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights. Users may not further distribute the material or use it for any profit-making activity or commercial gain.

Publisher permission
Permission for previously published items are in accordance with publisher's copyright policies sourced from the SHERPA RoMEO database. Links to full text versions (either Published or Post-print) are only available if corresponding publishers allow open access.

Take down policy
Contact lbscholars@cityu.edu.hk if you believe that this document breaches copyright and provide us with details. We will remove access to the work immediately and investigate your claim.

Download date: 20/11/2023
LEMON: a method to construct the local strains at horizontal gene transfer sites in gut metagenomics

Chen Li, Yiqi Jiang and Shuaicheng Li*

From Joint 30th International Conference on Genome Informatics (GIW) & Australian Bioinformatics and Computational Biology Society (ABACBS) Annual Conference
Sydney, Australia, 9-11 December 2019

Abstract

Background: Horizontal Gene Transfer (HGT) refers to the transfer of genetic materials between organisms through mechanisms other than parent-offspring inheritance. HGTs may affect human health through a large number of microorganisms, especially the gut microbiomes which the human body harbors. The transferred segments may lead to complicated local genome structural variations. Details of the local genome structure can elucidate the effects of the HGTs.

Results: In this work, we propose a graph-based method to reconstruct the local strains from the gut metagenomics data at the HGT sites. The method is implemented in a package named LEMON. The simulated results indicate that the method can identify transferred segments accurately on reference sequences of the microbiome. Simulation results illustrate that LEMON could recover local strains with complicated structure variation. Furthermore, the gene fusion points detected in real data near HGT breakpoints validate the accuracy of LEMON. Some strains reconstructed by LEMON have a replication time profile with lower standard error, which demonstrates HGT events recovered by LEMON is reliable.

Conclusions: Through LEMON we could reconstruct the sequence structure of bacteria, which harbors HGT events. This helps us to study gene flow among different microbial species.

Keywords: HGT, Local strain, Gut metagenomics, Graph

Recent research suggests that on average 81% of prokaryotes genes have been involved in HGT at some point in their history [4]. Their occurrences in multicellular eukaryotes are rare. However, several significant HGTs are still observed between bacteria and multicellular eukaryotes. Some are even common in specific environments. For example, we have observed HGTs from bacteria to fungi, from bacteria to the coffee borer beetle [5], as well as from virus, bacteria, and fungi to animals [6]. These recent discoveries have reshaped our understanding of evolutionary mechanisms.

HGTs may affect human health through a large number of human microbiota [7], including bacteria, fungi, archaea, and virus. They widely spread on human
biofluids and tissues, such as skin, lung, mouth. They are often associated with a range of human diseases and health conditions, from diabetes, colorectal cancer, to autism. The Human Microbiome Project [8] was launched in 2008 to study and understand the human microbiota. Some functions of the human microbiome, including antibiotic resistance and adaption to nutrients [9], are susceptible to HGT events. Mediated by phage, HGT in S. aureus occurs 1000 times more often than was thought, which greatly accelerates S. aureus to evolve resistance to antibiotics [10].

It is necessary to understand HGTs better. However, our current research is mainly focused on inferring ancient (lineage) HGTs from genomic sequences [11]. While the inference result is seriously affected by complex external factors [12]. For example, during the process of evolution, the transferred genome segments had been subjected to loss, mutation and duplication [13]. The inserted genes may also change the expression and functions of the gene around the insertion sites, resulting in very complicated structural variations [14], and temper with the receptor genome’s stabilities [15–17]. These possibilities complicate our detection of the HGTs. Better results can be achieved if we can anticipate these changes and correct for their effects. Recent efforts in human microbiomes provide us with such an opportunity.

LEMON(https://github.com/lichen2018/LEMON) takes use of existing shotgun NGS datasets to detect HGT breakpoints, identify the transferred segments, and reconstruct the local strain, which has complicated structure variation.

**Methods**

HGT events result in the integration of DNA segments from one species to another species, which will generate local strains containing DNA segments from different species. Figure 1 illustrates the workflow of LEMON.

![Fig. 1 The workflow of reconstructing HGT strains from pair-ends shotgun reads. Firstly, we map paired-end shotgun reads against reference genomes using Burrows Wheeler Aligner (BWA). Then, we select junction reads, whose two sides are mapped to two different references, from the set of mapped reads. Thirdly, by treating junction reads as points on a two-dimensional plane, we apply Density-Based Spatial Clustering of Applications with Noise (DBSCAN) to find candidate HGT breakpoints. Fourthly, we utilize split reads to get the exact positions of HGT breakpoints. Fifthly, according to the detected breakpoints, we could split references into segments, which are linked by junction reads. The coverage of each segment is calculated according to the number of mapped reads on it. Sixthly, we utilize the linked segments to construct a connected graph. By inserting dummy edges, we make the graph fully connected. Seventhly, we balance the coverage of each segment. Finally, we traverse the graph to find local strains. Each local strain should start from the first segment of one receptor and end with the last segment of the same receptor.](image-url)
References

Only the assembly results from multiple time-point metagenomics data of one individual can be used to discover the HGT events that exactly occurred between these time points. However, these samples are insufficient in published data, and this method cannot evaluate the difference in HGT events between different samples. To solve this problem, we construct a reference set $S$. We collected all of the bacterial genomes from the National Center for Biotechnology Information (NCBI) [18]. We selected one genome for each taxonomy as a representative genome to reduce the interference from homologous regions based on (1) the genome was annotated as reference or representative by NCBI; (2) or the one has minimal scaffolds number and highest completeness with contamination $<$10% in The Genome Taxonomy Database (GTDB) taxonomy evaluation results for 109,419 bacterial genomes [19]. The reference set contains 16,093 species with 1,246,881 scaffolds. Given a shotgun genomic read dataset $R$, we utilize BWA [20] to align reads against the references to identify the set of donors and receptors involved. These references with adequately covered segments are then retained. We denote the set of donors and receptors as $D$ and $H$ respectively.

Breakpoints and segments

The donor segments and receptor segments interweave in a local strain, separated by HGT breakpoints. We need to identify the breakpoints and segments involved in the local strains from $D$ and $H$. The data are heterogeneous. According to studies on virus integration [21] and studies on gut metagenomics, breakpoints are surrounded by mutations such as Single Nucleotide Variation (SNV), Copy Number Variation (CNV), short indels and inversions [22]. Hence we detect the breakpoints as follows.

First, we map paired-end shotgun reads to reference genomes using BWA, here all references are indexed together to generate Burrows–Wheeler Transform (BWT) indexes. If the two sides of a paired-end read are mapped to two different references, we call such a pair a junction pair, such as junction pairs $A$ and $B$ in Fig. 2a. The mapped positions of each junction pair give us two breakpoint candidates on the respective references. The two mapped positions of each junction pair can be treated as its coordinates on a two-dimensional plane. For example, $B_x$ and $B_y$ in Fig. 2b are mapped positions of junction pair $B$ on reference $X$ and $Y$ respectively. Then pair $B$ can be transformed to the point $B'$ with coordinates $(B_x, B_y)$ in Fig. 2c. All junction pairs mapped to

---

Fig. 2

(a) Some paired-end reads cross HGT breakpoints $s_1$ and $s_2$. (b) The two sides of one junction pair are mapped to two different references. $B_x$ and $B_y$ are mapped positions of junction pair $B$ on reference $X$ and $Y$ respectively. Furthermore, all junction pairs mapped to the same two references are transformed to points in a coordinate system with their mapped positions as coordinates, e.g. junction pairs $A$ and $B$ are transformed to points $A'(A_x, A_y)$ and $B'(B_x, B_y)$. (d) We apply DBSCAN using Euclidean distance to cluster the breakpoint pair candidates, such as $S_1$ and $S_2$. The maximum diameter of the cluster circle is the insert size of pair-end reads.
the same two references are transformed to points in a
coordinate system with their mapped positions as coordi-
mates. We then apply the clustering algorithm DBSCAN
[23] using Euclidean distance to cluster the breakpoint
pair candidates. A cluster that is supported by at least one
junction pair is further subjected to analysis to determine
the exact positions of its breakpoints. Next, we identify
the split reads which support a cluster. A read is split if it
can be partitioned into two parts, with each part mapped
to a different reference; we say it supports a cluster if
the mapped positions belong to the cluster. Each cluster
contains multiple breakpoint pair candidates.

To find the exact positions, we use a scoring scheme
to rank the candidate positions. The candidate with the
highest score is reported as the final breakpoint pair pos-
tion. The scoring scheme evaluates the split reads that
support the cluster. Suppose that there are two refer-
ences involved in the cluster, R1 and R2. Given a candidate
genomic positions pair p1 and p2 which belongs to R1 and
R2, respectively, we identify the split reads aligned to R1
where the alignment terminated at position p1. Denote
the portions of such a split read s mapped to R1 and R2
as e1(s) and e2(s), respectively. Then, the score is defined
according to the alignment qualities of e2(s) against R2
nearby p2. The alignment quality qr of e2(s) is calculated
as m/l, where m is the number of matched positions and
should be at least 15 bps, l is the length of e2(s). The qual-
ity score of p2 is calculated as 1 − ∑i log(1 − qr) of the
alignment qualities of the split read which supports the
respective cluster. Similarly we calculate the score for p1.
The candidate with the highest score is reported as the
final breakpoint pair positions of the cluster.

Each breakpoint involves two segments. Denote the seg-
ment pair as (eωu,vωv), and xωu,xωv ∈ {+, −} where + and
− respectively indicates the positive and negative strands,
and u,v ∈ V. We call such a pair a connected pair. A con-
ected pair is directed; that is, (vωu,uωv) ≠ (uωv,vωu).
Denote the set of connected pairs as E, the copy number of
e ∈ E as c(e) w.r.t. R. It is easy to see that the seg-
ments and connected pairs specify a graph G as illustrated
in Fig. 1.

Local strains
Assume that there are k HGT events captured in G. Since
each HGT event results in one local strain, there are
k local strains to be constructed according to G. The
first and last segments of each local strain are from the
same receptor. Without loss of generality let all the first
segments and the last segments of the local strains at
the integration sites be denoted B = { b1,...,bk1 }, and
T = { t1,...,tk2 }, respectively. Denote all the other segments
involved in the HGTs as V = { v1,...,vk } (excluding B and
T). Next, we estimate the number of copies (copy num-
ber) of each segment v within R, and denote it as c(v),
where v ∈ B ∪ T ∪ V. Factors such as species cover-
age are incorporated into the copy number estimation.
For example, if segment v1 is belong to a receptor r, the
coverage of v1 is cov(v1) and the average coverage of r is
cov(r), then the initial copy number of v1 is estimated as
c(v1) = cov(v1)/cov(r).

Our task is to identify the k local strains captured in G.
B and T can be identified with the input data. We assume
that the copy numbers of the first segment and the last
segment are the same for each strain, that is, c(b1) = c(t1),
without loss of generality.

Connectivity
We formulate the problem as a Eulerian circuit problem
to find the k local strains.

First, we insert dummy edges to transform the solu-
tion into a circuit. Without loss of generality, we assume
c(bi) = c(ti), 1 ≤ i ≤ k. We insert a dummy edge (t1+, b1−)
with the copy number c(bi) − 1, 1 ≤ i ≤ k. The edges
(t1+, b2−), 1 ≤ i ≤ k − 1, and edge (t2+, b1−) are inserted
with copy number 1.

Second, we insert edges and nodes to ensure connec-
tivity. In the ideal case, for each vertex v ∈ V, there
should be a path that starts from a source node in S, passes
through v, and ends up at some target in T. However,
due to sequencing errors, edges or nodes can be miss-
ing or spuriously introduced. If no target and source can
reach a node v, we remove v and its adjacent edges from
G. If a path exists from some vertex s to v, but no path
exists from v to a target t, we insert vertices and edges
to form a path from v to t. The inserted edge candidates
are taken from the reference sets D and S. If v belongs
to the same reference as t, this would suffice to recon-
struct the path. Otherwise, we add edges that connect
v and some nodes on the reference of t. In both situa-
tions, we use the minimum number of edges required.
All the introduced edges and vertex are assigned a copy
number of 1.

Balancing the graph
Denote the set of inbound edges of u as in(u) and the
outbound edges of u as out(u). That is, in(u) = { u|(v+, u−), (u−, v+) ∈ E, x ∈ {+, −} }, out(u) = { u|(v−, u+), (u+, v−) ∈ E, x ∈ {+, −} }. The in-copy and
out-copy of a vertex v are defined as cin(u) = ∑v∈in(u) c(e)
and cout(u) = ∑v∈out(u) c(e). In the ideal case, we should
have cin(u) = cout(u), but this may be broken due to experimen-
tal and sequencing errors.

We propose an integer linear programming (ILP)
approach to optimize the degree balance property. First,
assign each segment u (respectively μ) to a target copy
number t(u) (respectively t(μ)) according to Eq. 1c
(respectively 1d), to satisfy the degree balance property
(Eq. 1b). Then, the following program minimizes the disagreement between the assignment copy number and the target copy number (Eq. 1a).

minimize \( \sum u \epsilon_u + \sum \mu \epsilon_\mu \) (1a)

subject to \( c_{in}(u) = c(u), c_{out}(u) = c(u), \forall u \in S \cup V \cup T \) (1b)
\[ -\epsilon_u \leq c(u) - t(u) \leq \epsilon_u, \forall u \in S \cup V \cup T \] (1c)
\[ -\epsilon_\mu \leq c(\mu) - t(\mu) \leq \epsilon_\mu, \forall \mu \in J \] (1d)
\[ \epsilon_v, \epsilon_\mu \in \mathbb{R}^+ \] (1e)
\[ t(u), t(\mu) \in \mathcal{I}, t(u), t(\mu) \geq 1, t(\mu) \geq 1 \] (1f)

Finding Eulerian circuit
It can be shown that a Eulerian circuit to the graph constructed as illustrated in Fig. 1 gives a solution to our local strain reconstruction problem. However, the problem may yield multiple solutions. Each local strain should start from the first segment of one receptor and end with the last segment of the same receptor as illustrated in Fig. 1. Each reconstructed strain may contain several HGT events. Let the number of HGT events that contributes to the local strains \( i \) be denoted as \( h_i \). We choose a solution in which each reconstructed strain \( i \) has \( h_i \) as large as possible.

Evaluation metrics
To evaluate the performance of LEMON, we construct true local strains containing transferred segments and use LEMON to recover them. We propose two metrics Reconstruction Accuracy and Detection Rate to measure results.

We denote the true local strains as \( \{ H_i \}_{i=1}^n \), where \( n \) is the number of receptors, and each true local strain \( H_i \) consists of \( m_i \) segments, that is, \( H_i = \{ s_j^{H_i} \}_{j=1}^{m_i} \). We take the simulated reads as input of LEMON and construct reconstructed local strains \( \{ h_i \}_{i=1}^n \), where \( h_i \) is the reconstructed local strain which has the same receptor of \( H_i \); if \( H_i \) doesn’t have \( h_i \) in the result, we set \( h_i = \emptyset \). The segments in \( h_i \) are denoted \( s_j^h \), that is, \( h_i = \{ s_j^h \}_{j=1}^i \). We apply Smith–Waterman algorithm to measure the similarity between segment sequences of \( H_i \) and \( h_i \). \( \forall s_j^{H_i} \in h_i \) and \( \forall s_j^{H_i} \in H_i \), we consider \( s_j^{H_i} \) and \( s_j^h \) to be matched if and only if the breakpoint pair positions of \( s_j^{H_i} \) are both located within 20 bp around the breakpoint pair positions of \( s_j^h \). In our experiments, the parameters match_score and mismatch_score of Smith–Waterman algorithm are set as 1 and -1, respectively.

The reconstruction accuracy \( RA_i \) of \( h_i \) is defined in formula (2),

\[
RA_i = \frac{\text{SW}(h_i, H_i)}{m_i \cdot \text{match_score}, h_i \neq \emptyset} = 0, \quad h_i = \emptyset.
\] (2)

Here, \( \text{SW}(h_i, H_i) \) is the alignment score of \( h_i \) and \( H_i \), while \( m_i \) is the number of segments in \( H_i \). When all segments are matched in \( h_i \) and \( H_i \), which means all transferred segments are recovered. So \( \text{SW}(h_i, H_i) \) is equal to \( m_i \cdot \text{match_score} \), that is \( RA_i = 1 \).

We set the number of repetitions \( N \) as 8 and define the mean value of reconstruction accuracy \( \bar{RA} \) as follows,

\[
\bar{RA} = \frac{1}{N} \sum_{k=1}^{N} \frac{\sum_{i=1}^{N} RA_{ik}}{n_k}
\] (3)

\( n_k \) denotes the number of non-empty \( h_i \) in the \( j \)-th repetition.

An acceptable detection of one transferred segment should have its breakpoint pair located within 20 bp [24] of the true breakpoint pair as mentioned in “Breakpoints and segments” section. The Detection Rate is defined as the rate of the number of acceptable recovered transferred segments and the number of all transferred segments. The average detection rate is the average of 8 repetitions for each test.

Software parameter setting
Most third-party tools used in this article are set default parameters, including BWA, LUMPY [25], iRep [26], and STAR-Fusion [27]. The parameters of DBSCAN are \( eps \), which is the maximum radius of one cluster, and \( min_sample \), which is the minimum number of points in one cluster. In our paper, \( eps \) is set as the average insert size. \( min_sample \) is set as 1.

Results
HGT events detection in simulated human gut microbiomes
To simulate human gut microbiome with different complexity as mentioned in “Local strains” section, we constructed 5 simulated microbiomes containing 160, 320, 640, 1280, 2560 species, respectively. For each simulated microbiome, 5 different amounts (20, 40, 60, 80, 100) of HGT events were generated. For each HGT event, we randomly selected two reference sequences as the receptor and donor respectively. On the donor, we randomly selected one 10k bp sequence region as a transferred segment and inserted it to a randomly selected insertion position on the receptor. In this simulation, each HGT event contained one transferred segment. We denoted the
new receptor sequence containing transferred segments as the true local strain. All true local strains were used to generated 20X paired-end reads with WGSIM [28] as an input of LEMON. Eight repetitions were performed for every test.

In order to prove the performance of LEMON, we compared its performance with another popular breakpoint detection-based structural variant discovery software LUMPY.

Figure 3 illustrates a Comparison of Reconstruction Accuracy and Detection Rate between LEMON and LUMPY under different simulated conditions. The red dot in each boxplot denotes the mean value. As we can see, most mean values of Reconstruction Accuracy and Detection Rate achieved by LEMON are higher than those achieved by LUMPY, which demonstrates that LEMON can reconstruct more accurate strains and detect more transferred segments than LUMPY.

HGT breakpoints detection

In order to evaluate the performance of LEMON in HGT breakpoints detection as mentioned in “Breakpoints and segments” section, we applied it to local strains with different coverage and compared the performance with LUMPY. HGT events with random receptors, donors and breakpoints were generated in 100 randomly selected microbials, resulting in 60 local strains with 4260 HGT breakpoints. The HGT breakpoint is the insertion position of donor segments such as $s_1$ and $s_2$ in Fig. 2a. The paired-end reads generated from these local strains with 10 different values (2X, 5X, 10X, 15X, 20X, 30X, 40X, 50X, 60X, and 70X) of depth were input of LEMON and LUMPY. The performance is measured in terms of Sensitivity and False Discovery Rate (FDR) in breakpoints detection, and the bearing bias is 20 bp. If the distance between the detected breakpoint position and the true position is larger than 20 bp, we treat this detected position as one false detected position. Then if the distance is less than 20 bp, the detected position is treated as one true detected position. Therefore, FDR is the rate of false discovered positions among all discovered breakpoint positions. Sensitivity is the rate of true detected positions among all true breakpoint positions. LEMON has higher sensitivity and lower FDR than LUMPY across different coverage levels as illustrated in Fig. 4. At low coverage level, e.g. 2X and 5X coverage, LEMON can detect 22.39% and 51.19% of all HGT breakpoints, whereas LUMPY can detect 6.86% and 31.12% of all HGT breakpoints. At a higher coverage level, LEMON remains slightly better than LUMPY. For example, from 10X to 70X coverage, the detection sensitivity of LEMON ranges from 79.03 to 94.79%, whereas the detection sensitivity of LUMPY ranges from 70.06 to 93.47%. LEMON has lower FDR than LUMPY across different coverage levels, for example, at 2X, the FDR of LEMON is 0.002, while the FDR of LUMPY is 0.016. At 70X, the FDR of LEMON and LUMPY are 0.0034 and 0.010 respectively. Therefore, LEMON can detect more accurate breakpoints than LUMPY.

HGT strains reconstruction with complicated HGT event structure

In this simulation, we set the number of species $s$ to 2560 and the number of HGT events to 100. In order to simulate a complicated HGT event structure, we changed the number of transferred segments in each HGT event from 1 to 5. Simulated paired-end reads were generated from
true local strains by using WGSIM at 5X, 10X, 20X, and 30X coverage. 4 repetitions were performed.

Figure 5 shows the Comparison of Reconstruction Accuracy between LEMON and LUMPY under different coverage and number of transferred segments in one HGT event. We use $\Delta RA = RA_{LEMON} - RA_{LUMPY}$ to measure the performance difference between LEMON and LUMPY. If $\Delta RA$ is positive, LEMON achieves a higher $RA$ than LUMPY. As we can see from Fig. 5a, all $\Delta RA$ are positive and $\Delta RA$ decreases as coverage increases.

Figure 5b demonstrates that under different complexity of HGT event, $RA_{LEMON}$ is better than $RA_{LUMPY}$. Figure 5c gives an example of one HGT event containing a different number of transferred segments. Figure 5d and e demonstrate that LEMON has better performance than LUMPY across different levels of coverage, especially at low coverage levels such as 5X and 10X.

In Fig. 6, we compare local strains reconstructed by LEMON and LUMPY at 5X and 10X coverage levels. The true local strain contains two HGT events A and B. Each
HGT event has five transferred segments. +/- in each segment represents the forward/reverse direction of the segment. At 5X, LEMON detects one transferred segment Green-D6, while LUMPY fails to detect anyone transferred segment. At 30X, LEMON detects all transferred segment and has reconstructed the same structure as the true local strain, while LUMPY fails to detect Red +D1.

## Highly complex HGT structures do exist in real metagenomic data

We applied LEMON on a recently released metagenomic dataset [29] to reconstruct local strains containing HGT events. Some reconstructed local strains have complex HGT events, such as the reconstructed local strain of NZ_DS990133.1 in sample F-5 as shown in Fig. 7.

As we can see from Fig. 8, segments from one donor are not always inserted into the receptor as a whole. Sometimes they are inserted into the receptor together with segments from other donors. For example, Segments D1-D2-D3 and D2-D3-D4 from NZ_GG703855.1...
are inserted at the position of 181,680 bp and 431,930 bp on NZ_DS990133.1 respectively. The D2-D3-D4 from NZ_GG703855.1 together with D7 from NZ_GG703852.1 is inserted at the position of 433,142bp on the receptor. And the D8 from NZ_GG703852.1 together with D9 from NZ_GG703854.1 is reverse inserted at the position of 450,728bp on the receptor, which demonstrates the complexity of HGT events.

Local strains reconstructed by LEMON can assist replication timing profile restoring

We used iRep to estimate the replication timing profile of each bacterium in metagenomics data [29]. iRep utilizes linear regression to evaluate the coverage distribution across the genome to determine the PTR (peak-to-trough ratio), which is the ratio between the coverage at the origin and terminus of replication. However, due to the limitations of the reference sequence and the low sequencing depth of most species, we typically got very few replication timing profiles in a single metagenomics sample.

We applied iRep to evaluate two coverage distributions for each receptor. The first coverage distribution is evaluated based on the original reference of the receptor containing no HGT event. The second coverage distribution is evaluated based on the reconstructed reference of the receptor containing HGT events. According to the two coverage distributions, we estimated two replication timing profiles (including PTR value, predicted origin, and terminus position) for each receptor. Since iRep utilizes the regression method to estimate replication timing profiles, we use Standard Error to measure the accuracy of the estimated replication timing profiles. Figure 8 demonstrates the change of Standard Error at the origin and terminus of replication before and after reconstructions, some reconstructed local strains have much lower Standard Error, which means that LEMON help to reconstruct strains containing HGT events with more accurate restoring replication timing profile.

Verifying HGT breakpoints with gene fusion breakpoints detected from metatranscriptome data

In order to verify the HGT breakpoints detected by LEMON, we analyzed the IBD (Inflammatory Bowel

Table 1 Statistic table of HGT breakpoints type in strain results of eight samples

<table>
<thead>
<tr>
<th>Sample</th>
<th>Total breakpoints</th>
<th>Non-coding breakpoints</th>
<th>Candidate gene fusion points</th>
<th>Candidate gene fusion points ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>F-5</td>
<td>1024</td>
<td>227</td>
<td>228</td>
<td>22.3%</td>
</tr>
<tr>
<td>F-6</td>
<td>390</td>
<td>248</td>
<td>18</td>
<td>4.6%</td>
</tr>
<tr>
<td>F-7</td>
<td>437</td>
<td>261</td>
<td>24</td>
<td>5.5%</td>
</tr>
<tr>
<td>F-8</td>
<td>575</td>
<td>322</td>
<td>28</td>
<td>4.9%</td>
</tr>
<tr>
<td>F-9</td>
<td>230</td>
<td>157</td>
<td>11</td>
<td>4.8%</td>
</tr>
<tr>
<td>F-12</td>
<td>459</td>
<td>326</td>
<td>11</td>
<td>2.4%</td>
</tr>
<tr>
<td>F-14</td>
<td>364</td>
<td>243</td>
<td>14</td>
<td>3.8%</td>
</tr>
<tr>
<td>F-15</td>
<td>377</td>
<td>230</td>
<td>19</td>
<td>5.0%</td>
</tr>
</tbody>
</table>
Disease) data set published by HMP (Human Microbiome Project) [30]. In addition to metagenomic sequencing data, some samples in this data set have corresponding metatranscriptome sequencing data. The HGT breakpoints on DNA should cause some gene fusions in RNA. We used STAR-Fusion, the current state-of-the-art tool in gene fusion detection, on the metatranscriptome data to obtain gene fusion results. These results were compared with the breakpoint results in the corresponding metagenomic data obtained by LEMON and LUMPY. Three HGT breakpoints that have close gene fusions results within 200 bp were found among 17 pairs of metagenomics and metatranscriptome data as illustrated in Table 1. HGT breakpoints detected by LEMON and LUMPY have different shift distances away from fusion points detected by STAR-Fusion. This may validate that some gene fusions in bacterial chromosomes are caused by HGT.

The reads supporting the breakpoint, NZ_DS981501.1:4185-NZ_CP015401.2:963348, are shown in Additional file 1. The shift distance between the HGT breakpoint and the breakpoint obtained by STAR-Fusion is 10 bps. The shift sequence regions on the two reference sequences, such as TAATGTTTAG and TAATGTTTCA in Additional file 1, are almost the same.

We identify 3 main reasons for discrepancies between STAR-Fusion-detected gene fusion breakpoints and our HGT breakpoints:

1) The results of STAR-fusion are based on STAR aligner, while our algorithm is based on BWA. STAR aligner and BWA employ different alignment algorithm, giving rise to different breakpoints results;
2) Limited sequencing data. The amount of metagenomics sequencing data in the IBD data set is around 5G per sample, and the amount of metatranscriptome data is 2G per sample. This is insufficient for the statistical significance required for finding all the breakpoints;
3) Based on our statistics in Table 2, most of HGT breakpoints occur in the non-coding region.

In summary, it is reasonable to find only 3 matching breakpoints in 17 pairs of data.

**Conclusions and discussion**

In this paper we present LEMON, a novel HGT discovery software that can detect HGT events and reconstruct strains containing multiple HGT events with complicated structural variation.

Using LEMON to reconstruct the sequence structure of bacteria allows us to study the metagenomics problem from the sequence level, thus no longer subjected to the comparison of abundance. For example, since HGT is the fundamental mechanism for the spread of antibiotic resistance in bacteria, by utilizing LEMON we could detect transferred Antibiotic Resistance Genes (ARG)[31], determine the corresponding donors and receptors, and reconstruct strains of receptors, which harbor the transferred ARG. Therefore, we could get a better understanding of the transfer mechanism of ARG among bacteria.

However, as the amount of sequencing data is generally insufficient for current metagenomics analysis, challenges remain in identifying the HGTs sensitively and accurately. This results in several shortcomings in LEMON. First, LEMON remains weak in finding HGT between the sequences that do not exist in the reference genome. Second, because we only consider the reads of unique mapping, the HGT on the repeat region cannot be identified.

At present, our reference set only contains the genome of bacteria. However, human gut microbiome also contains other microorganisms such as fungi and viruses. Therefore, a reference library that contains sequences of bacterial, viral and fungal more completely would be highly desirable for HGT analysis of the microbiome.

**Table 2** Detail on the breakpoints of three gene fusion results which have close HGT breakpoints (HGT breakpoints located in 200 bp upstream or downstream around gene fusion points)

<table>
<thead>
<tr>
<th>Sample</th>
<th>H4009C2</th>
<th>C3003C3</th>
<th>C3003C3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upstream</td>
<td>NZ_GG703852.1</td>
<td>NZ_ACEP01000191.1</td>
<td>NZ_DS981501.1</td>
</tr>
<tr>
<td>Downstream</td>
<td>NZ_JRNC01000070.1</td>
<td>NZ_ACEP01000074.1</td>
<td>NZ_CP015401.2</td>
</tr>
<tr>
<td>HGT</td>
<td>751372</td>
<td>25274</td>
<td>4185</td>
</tr>
<tr>
<td>Upstream breakpoint</td>
<td>LUMPY 751114</td>
<td>25275</td>
<td>4220</td>
</tr>
<tr>
<td>Downstream breakpoint</td>
<td>LUMPY 127</td>
<td>27558</td>
<td>963349</td>
</tr>
<tr>
<td>STAR-Fusion</td>
<td>751131</td>
<td>25282</td>
<td>4195</td>
</tr>
<tr>
<td>HGT</td>
<td>199</td>
<td>27544</td>
<td>963348</td>
</tr>
<tr>
<td>STAR-Fusion</td>
<td>223</td>
<td>27588</td>
<td>963357</td>
</tr>
<tr>
<td>Upstream gene name</td>
<td>NZ_GG703852.1_gene2906</td>
<td>NZ_ACEP01000074.1_gene1544</td>
<td>NZ_DS981501.1_gene822</td>
</tr>
<tr>
<td>Downstream gene name</td>
<td>NZ_JRNC01000070.1_gene2118</td>
<td>NZ_ACEP01000019.1_gene553</td>
<td>NZ_CP015401.2_gene767</td>
</tr>
</tbody>
</table>
**Supplementary information**

Supplementary information accompanies this paper at https://doi.org/10.1186/s12859-019-3301-8.

**Additional file 1:** Detailed reads mapping result at HGT breakpoints NZD3981501.1:14185→NZC_CP015401.2:963348. Top-left is upstream genome; top-right is downstream genome. STAR-Fusion determined genes around gene fusion points are annotated with border bar. Red lines represent breakpoints detected from metagenomics data with HGT algorithm. Blue lines represent breakpoints detected from metatranscriptome data with STAR-Fusion. In red rectangle, top sequence with base name is local strain constructed with HGT breakpoints information, and other color bars are metagenomics reads support breakpoints. In blue rectangle, top sequence with base name is local strain constructed with gene fusion breakpoints, other color bars are metatranscriptome reads support those breakpoints.

**Abbreviations**


**Acknowledgements**

Not applicable.

**About this supplement**

This article has been published as part of BMC Bioinformatics Volume 20 Supplement 23, 2019: Proceedings of the Joint International GIW & ABACBS-2019 Conference on Bioinformatics. The full contents of the supplement are available online at https://bmcbioinformatics.biomedcentral.com/articles/supplements/volume-20-supplement-23.

**Authors’ contributions**

SL conceived the project. SL and CL designed the algorithm. CL implemented the algorithm. CL and YJ performed the analyses. CL, YJ and SL evaluated the results, and wrote the manuscript. All authors reviewed the manuscript. All authors read and approved the final manuscript.

**Authors’ information**

Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong SAR

**Funding**

The work is supported by City University of Hong Kong (Project 7004886). Publication costs are funded by City University of Hong Kong (Project 7004886). The funding body did not play any role in the design of the study and collection, analysis, interpretation of data, and manuscript writing.

**Availability of data and materials**

The real metagenomics dataset for local strains reconstruction was deposited to Sequence Read Archive (BioProject: PRJNA393237). IBD dataset was published by HMV and available on the Sequence Read Archive (BioProject: PRJNA392980).

**Ethics approval and consent to participate**

Not applicable.

**Consent for publication**

Not applicable.

**Competing interests**

The authors declare that they have no competing interests.

**Received:** 20 November 2019  **Accepted:** 2 December 2019  **Published:** 27 December 2019

**References**


Publisher’s Note
Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.