BIOINFORMATICS APPLICATIONS NOTE Vol. 30 no. 9 2014, pages 1297–1299
doi:10.1093/bioinformatics/btu017

Genome analysis

Advance Access publication January 13, 2014

PanGP: A tool for quickly analyzing bacterial pan-genome profile

Yongbing Zhao[1,2], Xinmiao Jia[1,2], Junhui Yang[1,2], Yunchao Ling[1,2], Zhang Zhang[1], Jun Yu[1], Jiayan Wu[1,*] and Jingfa Xiao[1,*]

[1]CAS Key Laboratory of Genome Sciences and Information, Beijing Institute of Genomics, Chinese Academy of Sciences, Beijing 100101, People's Republic of China and [2]University of Chinese Academy of Sciences, Beijing 100049, People's Republic of China

Associate Editor: John Hancock

ABSTRACT

Summary: Pan-genome analyses have shed light on the dynamics and evolution of bacterial genome from the point of population. The explosive growth of bacterial genome sequence also brought an extremely big challenge to pan-genome profile analysis. We developed a tool, named PanGP, to complete pan-genome profile analysis for large-scale strains efficiently. PanGP has integrated two sampling algorithms, totally random (TR) and distance guide (DG). The DG algorithm drew sample strain combinations on the basis of genome diversity of bacterial population. The performance of these two algorithms have been evaluated on four bacteria populations with strain numbers varying from 30 to 200, and the DG algorithm exhibited overwhelming advantage on accuracy and stability than the TR algorithm.

Availability: PanGP was developed with a user-friendly graphic interface and it was available at http://PanGP.big.ac.cn.

Contact: xiaojingfa@big.ac.cn or wujy@big.ac.cn

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on November 27, 2013; revised on December 30, 2013; accepted on January 6, 2014

1 INTRODUCTION

Since the conception 'pan-genome' was introduced in 2005 (Medini *et al.*, 2005), pan-genome had shed light on the genomic diversity and evolution of bacterial genome. To make pan-genome analyses more expediently, several software tools were designed. Panseq could identify single nucleotide polymorphism on core genome and strain's specific region (Laing *et al.*, 2010), and PGAP was designed to perform five analytic functions with only one command (Zhao *et al.*, 2012). At the same time, a series of mathematic models had been proposed to characterize pan-genome profile of bacteria, such as the *Streptococcus agalactiae* pan-genome model (Tettelin *et al.*, 2005), the *Haemophilus influenzae* pan-genome model (Hogg *et al.*, 2007), heaps law model (Tettelin *et al.*, 2008) and infinitely many genes model (Baumdicker *et al.*, 2010). All current models could well depict the pan-genome profile of a launch of bacteria with their time complexity $O(n2^n)$, where n is the strain number. In this article, we developed two sampling algorithms to efficiently perform pan-genome profile analysis for larger scale genomes. These two algorithms were integrated in the software with a graphic interface, PanGP.

*To whom correspondence should be addressed.

2 METHODS

2.1 Test dataset

Gene clusters from seven bacterial populations were used to test the performance of sampling algorithms; the detailed strain list and gene clustering method were introduced in Supplementary Material.

2.2 Sampling algorithm

To analyse the pan-genome profile of N strains, the pan-genome size and core genome size of n strains ($1 \le n \le N$) would be calculated. If a population has N strains, there are C_N^n combinations. When C_N^n grows to a large number, two sampling algorithms, totally random (TR) and distance guide (DG), would be available to sample s (s: sample size) combinations consisted of n strains.

For the TR algorithm: when $C_N^n > s \times r$ (r: sample repeat times), randomly sample s non-redundant combinations from the total C_N^n combinations. The average pan-genome size and core genome size of these s non-redundant combinations were record as $\overline{G}_{pan}(n)$ and $\overline{G}_{core}(n)$.

For the DG algorithm: when $C_N^n > s \times r$, sample $s \times k$ (k: amplification coefficient, which could be modified to control sample size for evaluating the genome diversity of the total combinations) non-redundant combinations from the total C_N^n combinations. Calculate the Dev_geneCluster value of these $s \times k$ combinations, sort all the $s \times k$ combinations by their Dev_geneCluster value and sample s combinations from end to end with the same interval. The average pan-genome size and core genome size of these s combinations were record as $\overline{G}_{pan}(n)$ and $\overline{G}_{core}(n)$.

The above sample process would repeat r times for both TR and DG algorithms, and their average value ($\overline{\overline{G}}_{pan}(n)$ and $\overline{\overline{G}}_{core}(n)$) would be taken as the pan-genome size and core genome size of n strains, respectively.

2.3 Genome diversity characterizing models in the DG algorithm

In the DG algorithm, genome diversity was taken as a criterion to sample a certain number of strain combinations. To figure out a suitable model to characterize genome diversity, three models were proposed. In the following models, $\boldsymbol{G}(n)$ represents a combination consisted of n strains. i and j represent the ith and jth strain in $\boldsymbol{G}(n)$. In the population with N strains, there are C_N^n $\boldsymbol{G}(n)$ combinations.

Model A: Genome diversity was characterized by the discrepancy of evolutionary distance on phylogenetic tree. Branch length (B_{ij}) and node depth (D_{ij}) between the ith and jth strain on phylogenetic tree were calculated respectively (Calculating method for B_{ij} and D_{ij} were

introduced in Supplementary Fig. S1 in Supplementary Material). The average branch length ($B^{avg}(n)$) or node depth ($D^{avg}(n)$) of every strain pair in a given combination $\boldsymbol{G}(n)$ represented genome diversity of the combination $\boldsymbol{G}(n)$ and were calculated using the following formulas, respectively.

$$B^{avg}(n) = \frac{1}{C_n^2} \sum_{1 \le i < j \le n} B_{ij}$$

$$D^{avg}(n) = \frac{1}{C_n^2} \sum_{1 \le i < j \le n} D_{ij}$$

In model A, six methods were provided, which were introduced in Supplementary Material.

Model B: Genome diversity was characterized by the difference of each strain's gene number. For each combination $\boldsymbol{G}(n)$, the total gene number of all strains (marked as Sum_geneNum) and the variance of gene numbers among those strains (marked as Dev_geneNum) were calculated by two different formulas, shown as follows.

$$V_{Sum\_geneNum}(n) = \sum_{1 \le i \le n} g_i$$

$$V_{Dev\_geneNum}(n) = \left(\sum_{1 \le i \le n} \left(g_i - {}^{V_{Sum\_geneNum}(n)}\!/_{n} \right)^2 \right)^{1/2}$$

Note: $V_{Sum\_geneNum}(n)$ means the Sum_geneNum value; $V_{Dev\_geneNum}(n)$ means the Dev_geneNum value; g_i is the gene number of the ith strain.

Model C: Genome diversity was characterized by the discrepancy among gene clusters. For each combination $\boldsymbol{G}(n)$, the average number of the different gene clusters between two strains (marked as Dev_geneCluster) was calculated to represent genome diversity in a given combination $\boldsymbol{G}(n)$. The value of Dev_geneCluster ($V_{Dev\_geneCluster}(n)$) is calculated by the following formula.

$$V_{Dev\_geneCluster}(n) = \frac{1}{C_n^2} \sum_{1 \le i < j \le n} |C_{ij}|$$

Note: C_{ij} is the number of different gene clusters between the ith and jth strains.

2.4 Evaluate the performance of sampling algorithm

To evaluate the accuracy of sampling algorithm, the pan-genome size and core genome size of n ($1 \le n \le$ N) strains were calculated by TR and DG algorithms. Real data were calculated without sampling; amplification coefficient k was set to 4; and the simulation were carried out with the sample size set to 300, 500, 800, 1000, 2000, 3000, 5000, 8000 and 10 000. For each population, the simulation for two sampling algorithms was repeated 100 times respectively. The accuracy for every simulation result was measured by root mean square (RMS) value, which calculated the deviation between sample result ($\overline{\overline{G}}_{pan}^{simulation}(n)$ and $\overline{\overline{G}}_{core}^{simulation}(n)$) and real data ($G_{pan}^{real}(n)$ and $G_{core}^{real}(n)$) at each point.

$$RMS_{accuracy} = \left(\frac{1}{2N} \left(\sum_{n=1}^{N} \left(\overline{\overline{G}}_{core}^{simulation}(n) - G_{core}^{real}(n) \right)^2 + \sum_{n=1}^{N} \left(\overline{\overline{G}}_{pan}^{simulation}(n) - G_{pan}^{real}(n) \right)^2 \right) \right)^{1/2}$$

The stability for every simulation result was also measured by RMS value, which calculated the deviation between the ith simulation result ($\overline{\overline{G}}_{pan}^{simulation_i}(n)$ and $\overline{\overline{G}}_{core}^{simulation_i}(n)$), and the jth simulation result ($\overline{\overline{G}}_{pan}^{simulation_j}(n)$ and $\overline{\overline{G}}_{core}^{simulation_j}(n)$) at each point ($1 \le i < j \le 100$).

$$RMS_{stability}^{i,j} = \left(\frac{1}{2N} \left(\sum_{n=1}^{N} \left(\overline{\overline{G}}_{core}^{simulation_i}(n) - \overline{\overline{G}}_{core}^{simulation_j}(n) \right)^2 + \sum_{n=1}^{N} \left(\overline{\overline{G}}_{pan}^{simulation_i}(n) - \overline{\overline{G}}_{pan}^{simulation_j}(n) \right)^2 \right) \right)^{1/2}$$

2.5 Nonlinear fitting for the pan-genome profile

When the pan-genome size and core genome size of n strains ($1 \le n \le$ N) were available, a series of mathematics models (Supplementary Material) were used to depict pan-genome size, core genome size and new gene size.

3 RESULTS

The performance of sampling algorithms. It was found in Supplementary Fig. S1 that genome diversity, characterized by Dev_geneCluster method in Model C, exhibited strong correlation to pan-genome size, and difference in gene cluster number may primarily affect pan-genome size. Hence, we take Dev_geneCluster as the method to characterize genome diversity in the DG algorithm. The performance of two sampling algorithms was estimated from three aspects—accuracy, stability and time cost. Seen from Supplementary Figs S2–S4, the result from the DG algorithm was closer to the real data and more stable than the TR result with the same sample size. For the same species, population size almost did not affect the accuracy and stability of the two sampling algorithms. When the sample size was set to 500, the error proportion in the total gene clusters, either between the DG result and the real data or between any two sample results from the DG algorithm, were <0.1% in all test populations. Regarding time cost, the DG algorithm took slightly more time than the TR algorithm with the same sample size (Supplementary Table S1). When sample size was set to 500, the analysis for ≤50 genomes could be finished within 3 min.

Application for PanGP. Among pan-genome analysis tools, PGAP, a comprehensive pan-genome analysis pipeline with five modules, could only deal with small-scale genomes in the pan-genome profile analysis module, while PanGP was a highly efficient tool for large-scale bacterial pan-genome profile analysis with sampling algorithms. Besides TR and DG algorithms, another one named traverse all algorithm (TA algorithm) was also provided in PanGP. When TA algorithm was selected, the pan-genome profile would be analysed without sampling. PanGP required orthologs information as the input data, which could be generated by series of software, such as PGAP (Zhao *et al.*, 2012), OrthoMCL (Li *et al.*, 2003) and PanOCT (Fouts *et al.*, 2012), with the format introduced in the online manual. When sampling finished, the pan-genome profile would be present as curves images (shown as Supplementary Fig. S5). The curves images could be revised and exported via the user-friendly graphic interface, and non-linear fitting with three mathematic models was also available for pan-genome size, core genome size and new gene size. More usages about PanGP were provided in the online manual.

Funding: The National Basic Research and Development Program (973 Program; 2010CB126604 to J.X.); National Programs for High Technology Research and Development (863 Program; 2012AA020409), the Ministry of Science and Technology of People's Republic of China; Key Program of

Chinese Academy of Sciences grant (KSZD-EW-TZ-009-02 to J.X).

Conflict of Interest: none declared.

REFERENCES

Baumdicker,F. *et al*. (2010) The diversity of a distributed genome in bacterial populations. *Ann. Appl. Probab*., **20**, 1567–1606.

Fouts,D.E. *et al*. (2012) PanOCT: automated clustering of orthologs using conserved gene neighborhood for pan-genomic analysis of bacterial strains and closely related species. *Nucleic Acids Res*., **40**, e172.

Hogg,J.S. *et al*. (2007) Characterization and modeling of the Haemophilus influenzae core and supragenomes based on the complete genomic sequences of Rd and 12 clinical nontypeable strains. *Genome Biol*., **8**, R103.

Laing,C. *et al*. (2010) Pan-genome sequence analysis using Panseq: an online tool for the rapid analysis of core and accessory genomic regions. *BMC Bioinformatics*, **11**, 461.

Li,L. *et al*. (2003) OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome Res*., **13**, 2178–2189.

Medini,D. *et al*. (2005) The microbial pan-genome. *Curr. Opin. Genet. Dev*., **15**, 589–594.

Tettelin,H. *et al*. (2005) Genome analysis of multiple pathogenic isolates of Streptococcus agalactiae: implications for the microbial "pan-genome". *Proc. Natl Acad. Sci. USA*, **102**, 13950–13955.

Tettelin,H. *et al*. (2008) Comparative genomics: the bacterial pan-genome. *Curr. Opin. Microbiol*., **11**, 472–477.

Zhao,Y. *et al*. (2012) PGAP: pan-genomes analysis pipeline. *Bioinformatics*, **28**, 416–418.