ABSTRACT

Motivation: Analyses in comparative genomics often require non-redundant genome datasets. Eliminating redundancy is not as simple as keeping one strain for each named species because genomes might be redundant at a higher taxonomic level than that of species for some analyses, some strains with different species names can be as similar as most strains sharing a species name, while some strains sharing a species name can be so different that they should be put into different groups, some genomes lack a species name.

Results: We have implemented a method and web server that clusters a genome data set into groups of redundant genomes at different thresholds based on a few phylogenomic distance measures.

Availability: The web interface, similarity and distance data, as well as R-scripts can be accessed at: http://microbiome.wlu.ca/research/redundancy/

1 INTRODUCTION

Genomic databases contain several sequenced strains for species of high interest, while other species contain only one genome and no evolutionarily close relatives (Wu et al., 2009). These biases might result in different qualities when predicting such features as functional interactions by comparative genomics. The problem has often been solved by using a measure of phylogenetic distance either directly as a score (e.g. Overbeek et al., 1999), or for normalizing scores (e.g. Zheng et al. 2005). The problem has also been solved by choosing single representatives for particular clades (e.g. Moreno-Hagelsieb and Janga 2008). However, selecting representatives by species name can be problematic because strains of different species, for example *Escherichia coli* and *Shigella flexneri*, can be so evolutionarily close that they should share their names; while strains of single species such as *Prochlorococcus marinus* might be very distant to each other (van Passel et al., 2006; Moreno-Hagelsieb and Janga, 2008). Besides some sequenced genomes do not have a species name, and the level of acceptable similarity among genomes for some applications might be lower than the most common for species (Moreno-Hagelsieb and Janga, 2008).

The Genomic Similarity Score (*GSS*) and other phylogenomic measures have been shown to concur with more computationally-demanding phylogenetic distances (Kunin et al., 2005; Alcaraz et al., 2010). We have used *GSS* to produce datasets of non-redundant genomes maximizing the number of predicted interactions by phylogenetic profiles (Moreno-Hagelsieb and Janga, 2008), and other practical uses, such as displaying the relationship between conservation of gene order and genomic similarity (Janga and Moreno-Hagelsieb, 2004). The datasets were built by ordering the genomes available by importance (e.g. *Escherichia coli* K12 would take priority because of its annotation quality), then from largest to smallest. We would then take the first genome in the list, and put any genome with a *GSS* above a chosen threshold into the list of redundant genomes. In the present work, we tested instead two clustering methods to group genomes, besides a couple variants of the *GSS* and DNA signatures (Campbell et al., 1999). Since other research groups have asked us for lists of non-redundant genomes, here we make them available to the wider community.

2 DATA AND METHODS

We used 2160 prokaryotic genomes available at the RefSeq database (Pruitt et al., 2007) (ftp://ftp.ncbi.nih.gov/genomes/Bacteria/) by November 2012. Orthologs were reciprocal best hits (RBH) as described previously (Moreno-Hagelsieb and Latimer, 2008). We plan to update the data every six months.

We calculated the *GSS*s for each genome pair as follows (Janga and Moreno-Hagelsieb, 2004; Moreno-Hagelsieb and Janga, 2008):

\[ \text{GSS}_{ij} = \frac{\sum_{i=1}^{n} \text{compScore}_{ij}}{\sum_{i=1}^{n} \text{selfScore}_{ij}} \]

where *compScore* is the score of the alignment of protein *i* in genome *A* against its RBH in genome *B*, and *selfScore* is the score of the alignment of protein *i* against itself. Because the *selfScore* for proteins in genome *A* might differ from the *selfScore* for proteins in genome *B*, the final *GSS* is the average of the calculation from the perspective of both genomes.

We also calculated two modified version of the *GSS*:

\[ \text{GSS}_{B} = \frac{\sum_{i=1}^{n} \text{compScore}_{ij} \cdot \text{selfAlignLength}_{i}}{\sum_{i=1}^{n} \text{selfScore}_{ij} \cdot \text{selfAlignLength}_{i}} \]

Here the scores were normalized against the length of the alignments. Because *selfScores* normally come from longer alignments than *compScores*, *GSS* is larger than *GSS*._B_. The final *GSS*_A_ is the average of this calculation from the perspective of both genomes under analysis.

\[ \text{GSS}_{C} = \frac{\sum_{i=1}^{n} \text{compScore}_{ij} \cdot \text{selfAlignLength}_{i}}{\sum_{i=1}^{n} \text{selfScore}_{ij} \cdot \text{selfAlignLength}_{i}} \]

Instead of the *selfScores* of proteins finding RBH, we calculated the total sum of *selfScores* of all proteins in the genome of reference. This has the effect of producing a smaller *GSS*. Again, the final *GSS*_C_ was the average from the perspective of both genomes under analysis.

We also clustered our genomes by *dr*, *tri*-, and *tetra*-nucleotide DNA signatures with Manhattan distances (Campbell et al., 1999).
To build groups of redundant genomes, we used a hierarchical clustering method (Hastie et al., 2009). We used 1 – GSS to construct a dissimilarity matrix to run the clustering algorithms.

All computations were carried out using R (R Development Core Team, 2012). The commands agnes and diana in the cluster package were used to run the two main strategies in hierarchical clustering: agglomerative or bottom-up, and divisive or top-down, respectively.

To assess the clustering structures, we used the agglomerative/divisive coefficients defined as 1 minus the average ratio of dissimilarity of one unit to the cluster it first merges with, to the dissimilarity of its merging in the final step of the algorithm. A coefficient close to zero means that the algorithm did not find a natural cluster structure (the data consist of a single cluster), while a coefficient close to 1 means that very clear clusters have been identified.

3 RESULTS

$GSS_a$ and $GSS_b$ concentrate strains sharing species names (1103 strains in 224 species) into the 0.9 to 1.0 range of the GSS distribution, while $GSS_c$ spreads several of them between the 0.7 to 0.9 range (Fig. 1). Thus, $GSS_c$ might be the least adequate for clustering redundant genomes. Since DNA signatures are faster to calculate than $GSS_a$, we are starting to experiment with DNA signatures for our work in comparative genomics. Thus, we have also clustered our genome dataset based on di-, tri- and tetra-nucleotide signature distances. DNA signature distances among same-species strains concentrate at less than 0.03.

The divisive coefficients were higher, and thus better, than the agglomerative coefficients using all three $GSS_a$ and DNA signature distances (Fig 1). The coefficients using $GSS_a$ were the lowest among all three $GSS_a$ for both algorithms thus confirming that $GSS_a$ is the least adequate for clustering genomes. However, DNA signature distances produced the best overall clusters with divisive coefficients above 0.95.

The appropriate threshold for defining groups of redundant genomes depends on the application (e.g. Moreno-Hagelsieb and Janga 2008). Thus, we recommend experimenting with different thresholds. The web site provides ten thresholds per measure that inexperienced users would be able to modify these scripts to obtain groups at other thresholds. Finally, we also provide cluster files that can be read by software for drawing and manipulating phylogenetic trees (Newick format).

REFERENCES


