**Introduction**

A Single-cell Amplified Genome (SAG) is a sequenced genome of an individual cell. They can be sequenced because their DNA is amplified to sufficiently high levels. However, the amplification process results in incomplete coverage. By counting COGs [elaborate, cite!], we estimate many SAGs may only be 15% to 60% complete [histograms!]. We implement and evaluate a tool, Sagex (or SAG EXtrapolator), to remedy this issue via bioinformatics means. Sagex is a classifier that is trained on the incomplete SAG assembly and uses an assembled metagenome to classify metagenome contigs as members of the SAG’s population genome—a metagenome of the phylogenetically nearest members of the community to the SAG [correct??? Cite???].

Sagex’s classification errors are favourable. It is capable of being a high Positive Predictive Value (PPV) classifier, so the hits it calls are confident. It can also retain this high PPV without losing too much sensitivity. For example, while evaluating Sagex with EColi, Sagex was able to maintain 98% PPV and 60% sensitivity (verify!). Sagex is also capable of contributing to greater SAG population genome completeness by an estimated 23% on high-PPV settings. Trivially, SAGs can be extrapolated to near total completion on low-PPV settings.

**The Sagex pipeline**

Sagex (short for SAG Extrapolator) is a tool for classifying metagenomic contigs as having come from a SAG’s population genome. The population genome is the metagenome of an individual genome’s phylogenetically nearest members. One could say is it quite nearly (but not) the individual’s genome. The tool has a simple interface (see figure 1) with pre-tuned parameter defaults which may be changed if desired, allowing for Sagex to be useful to both novice and experienced users. In the simplest interface, Sagex accepts two FASTA files as inputs and provides one as an output. A novice user should try entering a single assembled SAG and assembled metagenome into Sagex to get a subset of the metagnome’s FASTA as an output.

RAID1:Users:wdurno:sagex:sagExtrap5:sagex:paper:simpleSagex.pdf

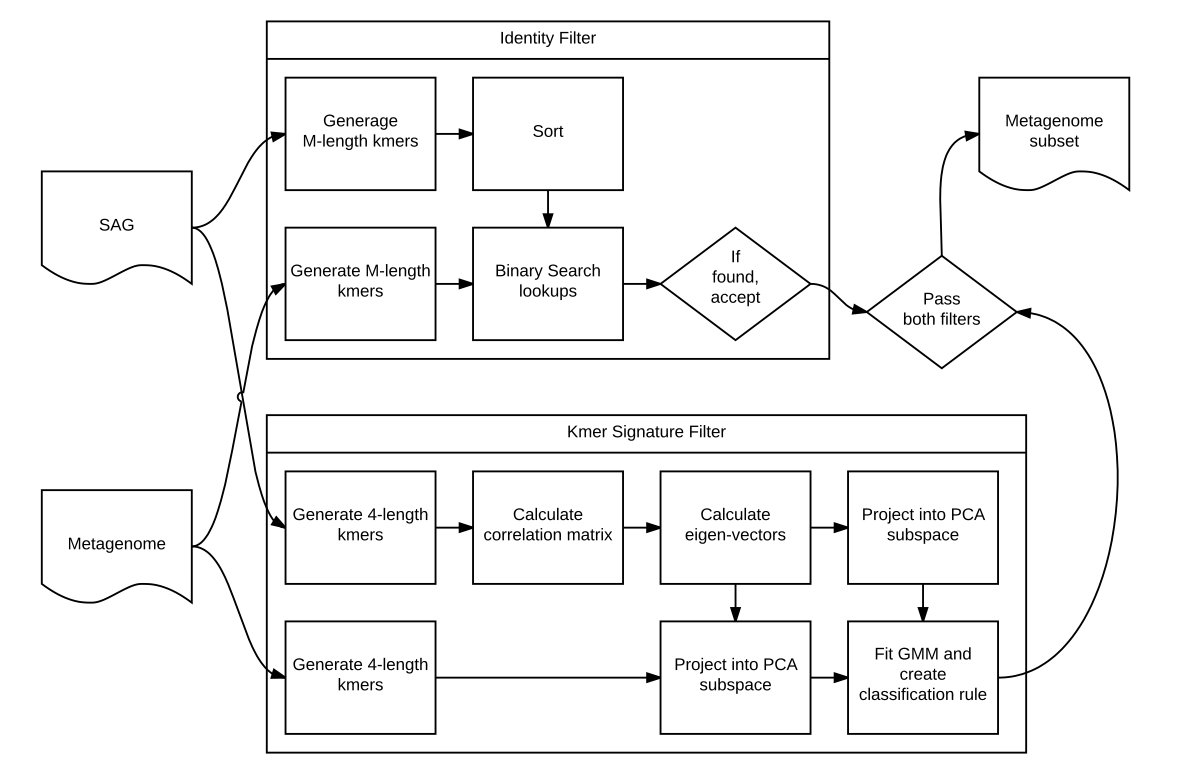
**Figure 1. Simple Sagex Pipeline**

Sagex can be as easy to use as entering a single SAG and metagenome FASTA to get an output FASTA which is a subset of the Metagenome FASTA. The output is the set of metagenome contigs classified to be shared with the SAG’s population genome.

Sagex can also be used in more advanced ways. To understand how, we now elaborate on the inner workings of the pipeline (see figure 2). The classification process is largely inspired by [1]. The classification rule for each metagenomic contig is that it must pass two tests: (1) it must pass an identity filter, and (2) it must have a similar kmer signature as the SAG. The identity filter requires that there be at least one contig in the SAG that shares a minimum number of contiguous nucleotides with it. An error analysis has shown that, for Sagex, a default of 21 is sufficient. The kmer signature of all contigs is a list of 256 tetranucleotide proportions by default. The kmer signature of the SAG is learned by performing a Principal Component Analysis (PCA) on the data and fitting a Gaussian Mixture Model (GMM) to the SAG (see figure 3). If the metagenomic contig falls within a 95% confidence region, the contig passes the test.

The identity filter module works by accepting a user-specific minimum contiguous agreement length of M. Sagex then orders all M-length substrings from the SAG FASTA so that it may perform binary searches for M-length substrings of metagenome contigs. If any match is found, the metagenomic contig passes the identity filter test.

The kmer signature module works by first calculating the tetranucleotide proportions of each contig. Then, the correlation matrix of the metagenomic kmers is used to generate eigenvectors. These eigenvectors are then used to project the kmers of the SAGs and metagenome into the PCA subspace of the metagenome. PCA is valuable because it reduces the number of parameters required to estimate in the following step. Next, the GMM will attempt to fit to the SAG via the expectation maximization algorithm initialized by k-means which, in turn, is initialized with Llyod’s algorithm [Cite!]. If the k-means estimation algorithm fails to fit a user-specified number of Gaussians to the data, Sagex will reduce the number of Gaussians and try again. The statistic used to classify the metagenomic contig is the minimum Euclidian distance to a Gaussian’s centroid. If there is only a single Gaussian in the GMM, then the distance is null-distributed according to a Chi-Squared distribution. If there are more Gaussians, then a null distribution is inferred by simulating values from the GMM. From this null distribution, a 95% confidence region may be found. If a metagenomic kmer falls within the region, we say it passes the kmer signature test.

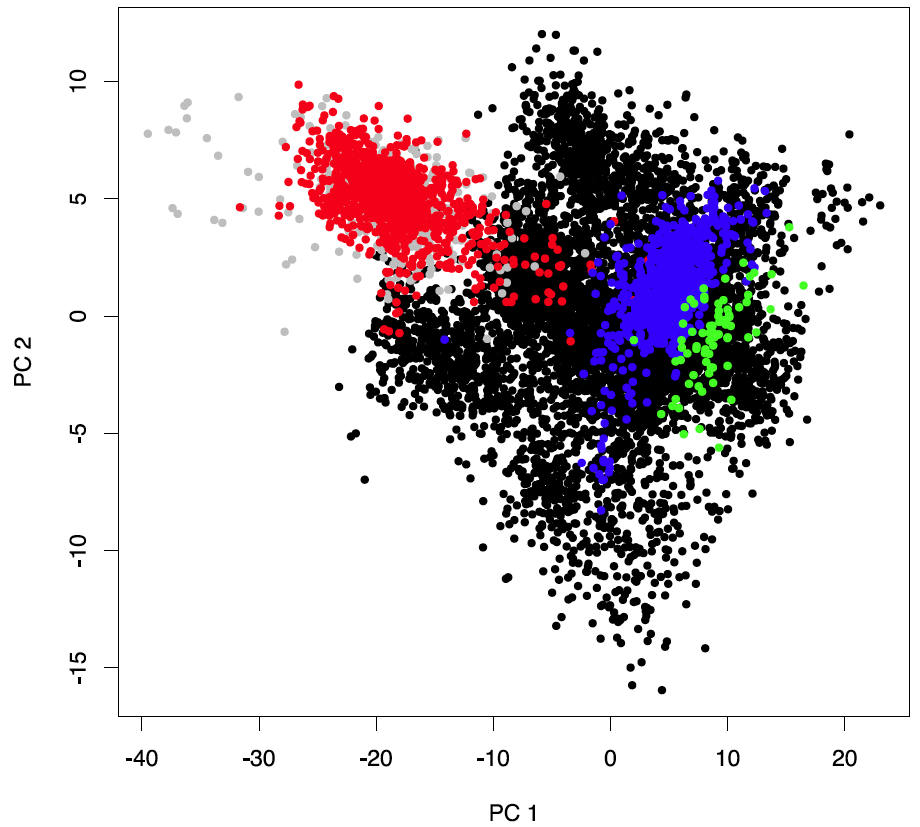


**Figure 2. Expanded Sagex pipeline**

The pipeline has two major components, the identity filter and the kmer signature filter.

Of all Sagex options, the kmer PCA feature is the most popular. Notice that the kmer signature filter must compute a kmer PCA. With the activation of a flag (-X “file destination path”), Sagex will write the kmer PCA to disk. It is prepared to be read into and visualized with R.

Sagex can also chop up contigs into equivalent lengths instead of using proportions. Because each contig may have several chops, an additional voting step is added to classification. If a minimum proportion of chops are classified as hits, the contig is counted as a hit.



**Figure 3. Kmer space signatures**

A variety of genomes’ kmer signatures are plotted. Each contig is a single point in 256 dimensions of tetranuculeotide frequency proportions. Here, two principal components are used to visualize the higher dimensional space. Green points belong to a Pelagibacter SAG. Blue points belong to a NITMA SAG. Red points belong to an EColi genome. Grey points belong to an EColi SAG. Notice that the genome and SAG overlap significantly. Black points belong to a Saanich metagenome and occur in several clusters which makes sense because of the many genomes which comprise it.

**Classification error analysis**

In the following analysis we see that Sagex is capable of being a high Positive Predictive Value (PPV) and high Negative Predictive Value (NPV) classifier without sacrificing too much sensitivity or specificity. That said, this does require that the metagenomic assembly does not have a high rate of chimeras (false contigs produced by assembling DNA from different organisms). As the quality of sequencing and assembling increases, so will the accuracy of this analysis.

To estimate the classification errors of Sagex, three genomes were used: (1. genome) a completely assembled EColi mg1655 genome[cite NCBI source!], (2. SAG) an EColi mg1655 SAG [2], and (3. metagenome) a Saanich Inlet metagenome from which we do not expect EColi mg1655 to occur. The SAG-genome pair was selected because it is a lab-cultivated strain and thus has minimal genome differences. The SAG was used to train Sagex. Contigs generated from the genome were labeled as correct hits. Contigs from the metagnome were labeled as false hits.

[1] Jeremy A. Dodsworth et al. “Single-cell and metagenomic analyses indicate a fermentative and saccharolytic lifestyle for members of the OP9 lineage” *Nature Communications* **4**, Article number: 1854 doi:10.1038/ncomms2884. Received 11 November 2012. Accepted 13 April 2013. Published 14 May 2013

[2] Jeff Gole et al. “Massively parallel polymerase cloning and genome sequencing of single cells using nanoliter microwells” *Nature Biotechnology* **31**, 1126–1132 (2013) doi:10.1038/nbt.2720 Received 23 April 2013 Accepted 13 September 2013 Published online 10 November 2013