MetaMutationalSigs

Abstract:

Motivation:

Mutational signature analysis is a very active and important area of interest. There are several packages available now for mutational signature analysis and they all use different approaches and give nontrivially different results. Because of the differences in their results, it is important for researchers to survey the available tools and make choose the one that best suits their application. There is a need for software that can aggregate the results from different packages and present them in a user-friendly way to facilitate effective comparison.

Results:

We created this package *MetaMutationalSigs* to facilitate comprehensive mutational signature analysis by creating a wrapper for different packages and providing a standard format for their outputs so that they can be effectively compared. We have also standardized the input formats accepted by various packages to ease interoperability. We also create standard visualizations for the results of all packages to ensure easy analysis. Our software is easy to install and use through Docker, a package manager that automates the dependencies.

Introduction:

The basic idea behind mutational signatures is that mutational processes create specific patterns of mutations. Thus it follows that if one can identify these patterns in a given sample then they can essentially detect the corresponding mutational processes. Since there are a huge variety of mutations possible, they are grouped into 6 major types based on the base where the mutation was observed. These 6 mutation types are C>A, C>G, C>T, T>A, T>C, and T>G. Now, these 6 types of mutations are further divided based on their location, i.e. other bases that are in their immediate proximity giving us the 96 mutation types that are termed the single based substitution context.

The mutational signature analysis workflow involves multiple steps that require different amounts of time and processing power. Typically, one starts with BAM files that are aligned to some reference genome and then proceeds to the variant calling step which outputs VCF files. These steps are usually very resource-intensive and thus do not allow for much experimentation, the downstream steps of variant filtering and annotation are much less expensive. The final step

is the actual mutational signature analysis, which is the least resource-intensive and thus allows for experimentation with different methods.

We created this package MetaMutationalSigs to facilitate comprehensive mutational signature analysis by creating a wrapper for different packages and providing a standard format for their outputs so that they can be effectively compared. We have also standardized the input formats accepted by various packages to ease interoperability. We also create standard visualizations for the results of all packages to ensure easy analysis. Our software is easy to install and use through Docker.

Approach:

There are two different methods usually used for mutational signature analysis: signature refitting, and de-novo signature extraction. Signature refitting methods try to recreate the observed mutational pattern in the sample (the frequencies of 96 types of mutations) using the linear combinations of known signatures (COSMIC 30, SBS, ID, etc.), these methods work quite well on small sample sizes and single samples and are widely used as such. Signature extraction methods try to find new signatures from a given dataset using a set of samples, the newly extracted signatures are then compared with known reference signatures and the novel signature is assigned to a known signature if their cosine similarity exceeds a set threshold. There are a few important caveats to signature extraction: 1) a novel signature can be very similar to several reference signatures and the assignment is not always perfect 2) the threshold for assignment plays a crucial role but is not widely agreed upon, using a different threshold can change the assignment.

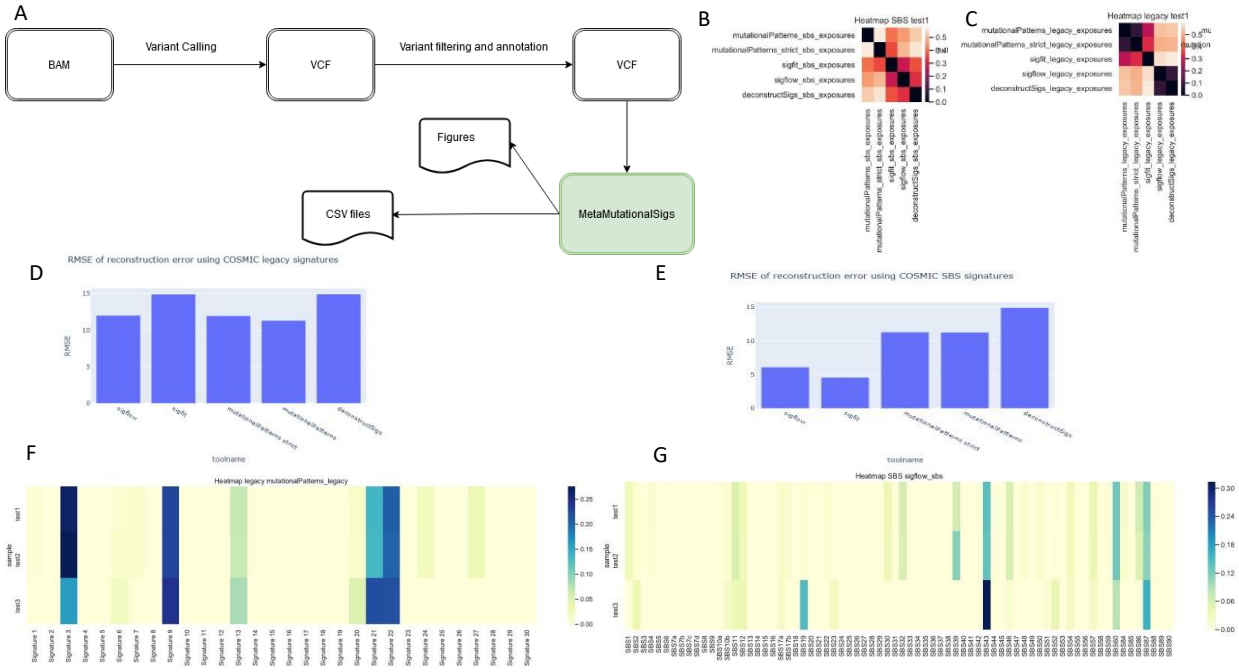
We chose signature refitting as our primary task and implemented high performing packages as identified in () that were implemented in R. Our package outputs several data files in CSV format ready for further analysis and visualization using external packages along with visualizations of the signature exposures as described in table 1.

We use the RMSE of the reconstruction error as our performance metric for comparison of these packages.

Table 1.

File Name	Format	Description
Heatmap_exposures_all_sigs_legacy.pdf	pdf	Exposures for all COSMIC 30 signatures.
Heatmap_exposures_all_sigs_SBS.pdf	pdf	Exposures for all COSMIC SBS signatures.
Heatmap_legacy.pdf	pdf	Heatmap for difference between the predicted exposures by different tools. One for each sample.
Heatmap_SBS.pdf	pdf	Heatmap for difference between the predicted exposures by different tools. One for each sample.
legacy_pie_charts.html	html	Interactive pie charts of 30 legacy signature exposures, per sample and for each tool.
sbs_pie_charts.html	html	Interactive pie charts of SBS signature exposures, per sample and for each tool.
legacy_rmse_bar_plot.png	png	Reconstruction error using 30 legacy COSMIC signatures for each tool.
sbs_rmse_bar_plot.png	png	Reconstruction error using COSMIC SBS signatures for each tool.
toolname_results\legacy_sample_error.csv	csv	Data used to create the bar plot.
toolname_results\legacy_sample_exposure.csv	csv	data used to create heatmap and pie chart.
toolname_results\sbs_sample_error.csv	csv	Data used to create the bar plot.
toolname_results\sbs_sample_exposure.csv	csv	data used to create heatmap and pie chart.

Figure 1



A) Workflow for mutational signature analysis. Our tool MetaMutationalSigs is at the final level of analysis. B) Heatmap of Euclidean distance between the predicted exposures of COSMIC legacy signatures by different tools for a sample. C) Heatmap of Euclidean distance between the predicted exposures of COSMIC SBS signatures by different tools for a sample. D) RMSE of the reconstruction error using COSMIC legacy signatures for each tool, lower is better. E) RMSE of the reconstruction error using COSMIC SBS signatures for each tool, lower is better. F) Heatmap of COSMIC legacy signature exposures, one row per sample. G) Heatmap of COSMIC SBS signature exposures, one row per sample.

Discussion:

The massive increase in the number of software packages has made managing dependencies quite burdensome, coupled with incompatible data formats for signature matrices can make mutational signature analysis difficult and hard to reproduce. Our package provides an easy way of performing these setup related tasks so one can focus more on the analysis. Future work for this project would focus on expanding the tool to work with more packages and keep the reference signatures updated as new versions are released. Due to the open-source nature of the project, we also welcome additional feature requests using the project link on GitHub https://github.com/PalashPandey/MetaMutationalSigs