Introduction

Potential breakthroughs in cancer therapy success could come from personalised cancer therapy, based on somatic mutations, particularly Single Nucleotide Polymorphisms or Variants (SNPs or SNVs) in an individual. Many tools have been developed to differentiate somatic mutations from germline. These variant callers take into account different parameters and vary in performance. Mutect2, Freebayes, VarScan and VarDict are variant calling tools developed to identify somatic variants. However, the performance and results obtained, such as the precision and recall, differ from one variant caller to another due to the differences in their workflow. Their performance can also vary depending on the sample. Among the four variant callers, Mutect2 is more conservative and has high precision. While VarScan does not have as high precision as Mutect2, it is more liberal in identifying somatic mutation and excels in discovering somatic SNVs with high allele frequency. Mutect2, Freebayes and VarScan use Bayesian statistics to determine whether a variant is somatic, whereas VarDict uses read alignment. Ensemble-based approaches, which integrate calling results from multiple somatic variant callers, have been found to improve the identification of somatic variants over individual callers in the ICGC-TCGA Dream Mutation Calling challenge **[1].** This project aims to construct a meta-approach that integrates the calls from the various variant callers that can outperform the individual variant callers.

Methods

As a baseline, we considered a naive approach, which only called a variant if it was called by at least two of the four variant callers, and a supervised machine learning approach to build a decision tree-based classifier, specifically C4.5. We were limited to only using features that were already available in the output files from the four callers to train our classifier (Figure 1).

To resolve the disagreement between the variants detected between the two platforms, we developed an ensemble method that combines the outputs from the four variant callers, which we hoped will inherit the high precision of Mutect2 and high sensitivity of Freebayes and Vardict. Previous studies have found that ML-based ensemble approaches are more powerful than ML-based callers from pure genomic features (such as Snooper10 and MutationSeq11) since metric values produced from callers are incorporated **[2].**

To allow our classifier to predict somatic SNPs, our classifier should be trained using only features that associate well with class labels. We compiled all loci that were identified by the four variant callers and merged them with the truth labels from the bed file for feature evaluation by supervised learning in Weka. For each variant caller, features that associate most highly with a true positive call were identified using the Chi-square test. Then, for each variant, the call from each of the four callers was added as features (i.e. FILTER_Freebayes is a feature with the value indicating whether a variant was identified as SNP by freebayes). All features containing string values were converted into categorical features or removed. To reduce the complexity of the model and speed up training, duplicate features and features that associate poorly with the class labels were removed. The final feature set preserved for further analysis combined the top features ranked by the Chi-square test and the FILTER features (Table A1). The feature selection results illustrated that somatic callers not only predict variants but provide valuable measurements related to somatic variant prediction as well.

It is known that the success of deep learning approaches fluctuates depending on the training dataset. To produce the training data for our model, the 4 VCF files from each folder were merged into the ARFF file in R which contains the top 16 attributes and the truth label. These ARFF files were then combined in Weka. Due to the nature of the workflow, many instances contained missing data for attributes. For example, variants considered by Freebayes but not by Mutect2 would have missing values in attributes contributed by the Mutect2 file. To handle missing values, several methods were considered: assigning with a constant value (such as -1, 0, 1), assigning with the mean value of the column, or distributing the instance to both child nodes with diminished weights. We found the last method using fractional instances vastly outperformed the rest (Figure A2a). In Weka, C4.5 is implemented as the J48 classifier and can deal with missing data as we had intended by creating fractional instances at internal nodes. J48 is a decision tree classification algorithm based on C4.5 which was chosen for our model. Briefly, having "small" leaves increases the chance that the decision tree is overfitting to the test dataset, whereas leaves that are too "large" may result in underfitting and thus an inaccurate model. To reduce the risk of overfitting, the decision tree was pruned to remove internal nodes that created leaf nodes that did not contain a specified number of instances. We varied this number to identify the minimum size that gives a reliable F1 score (Figure A2b).

We wanted to determine whether a subset of the full attribute list can yield better results. Therefore, the Chi-squared test was performed to rank the attributes. 5-fold cross-validation was performed for the top 100%, 75% and 50% of the full attribute list and the precision, recall and F1 score was compared. When the feature list was pruned, the performance of our model diminished rapidly (Figure A3), i.e. it is unlikely that any of the attributes were redundant.

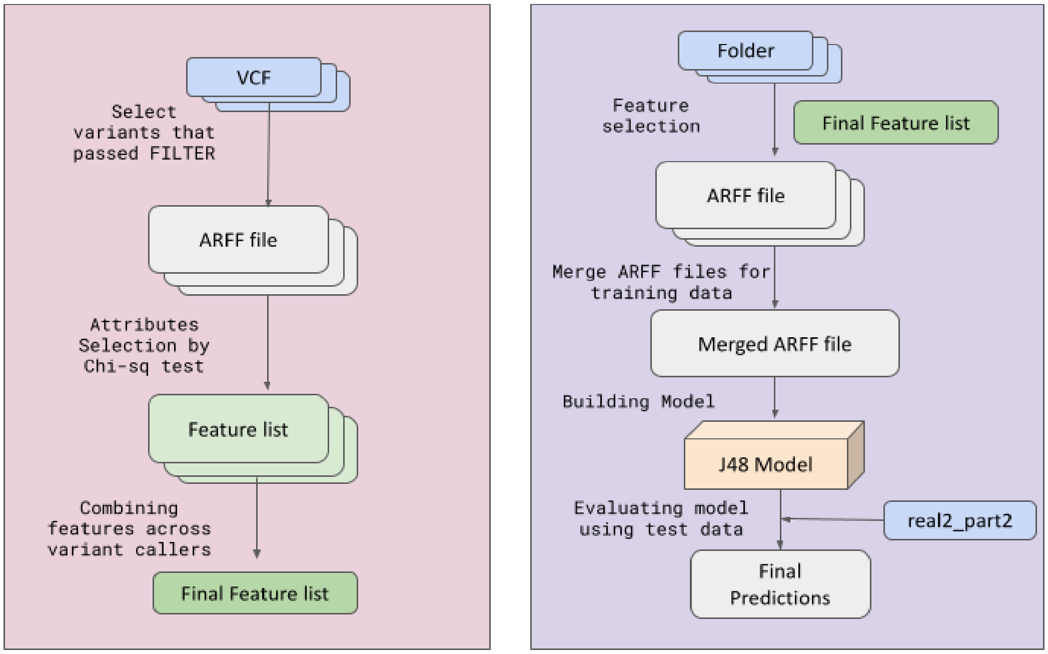
VCF
Select variants that passed FILTER
ARFF file
Attributes Selection by Chi-sq test
Feature list
Combining features across variant callers
Final Feature list
Folder
Feature selection
Final Feature list
ARFF file
Merge ARFF files for training data
Merged ARFF file
Building Model
J48 Model
Evaluating model using test data
real2_part2
Final Predictions

Figure 1. *Workflow for the Supervised Machine Learning Approach*.

Results

The naive approach identified variants by checking that it was called by at least two out of the four variant callers. However, its poor performance makes it evident that the naive method is unreliable (Table A2). The deep learning approach yielded better results for all test datasets as compared to the naive method (Figure 1, Table A3). To verify that the method was reliable for any subset of the data, we constructed a combined training dataset with the datasets provided except a pre-defined test dataset. This was done for each dataset from real1 to syn5. Overall, our method outperformed the naive method for every sample (Table A3). Using the decision tree constructed from syn1-5 and real1, we tested it on the real2_part1 dataset and found similar results (Table A4), indicating that our model is applicable to diverse sets of data.

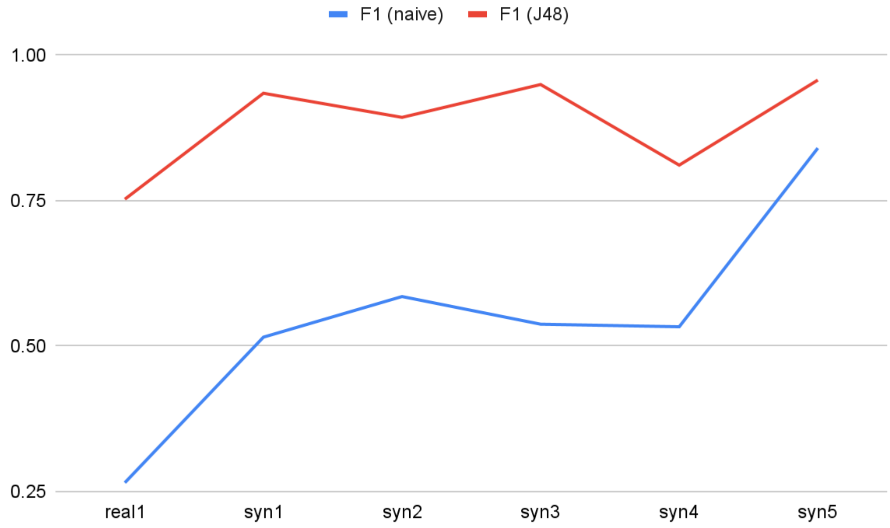

Figure 1. *<u>The deep approach consistently outperforms the naive method</u>. F1 scores yielded from J48 are higher compared to the naive method when tested on each dataset from syn1-5 and real1*

Conclusion

To ensure our machine learning approach was trained with a large dataset, we combined the datasets of real1 and syn1-5 to generate a single large dataset. Approaches using synthetic data also rely on the assumption that the mechanism for somatic mutations by cancer can be accurately modelled to generate synthetic variants. As synthetic datasets constitute the bulk of our training data, the classifier can only be used for real cancer samples if the distribution of somatic mutations in the synthetic dataset follows that of mutations in real cancer samples. This is a reasonable assumption: DREAM samples have mutations spiked into BAM files by modifying reads covering selected sites, realigning a specified number to obtain the desired VAF, and replacing the original reads to create realistic tumour samples. In our approach, instances correctly identifying non-SNPs were discarded to reduce the cardinality of the dataset. This means that features that associate well with true negatives are not considered during feature evaluation. Lastly, since the best performing attributes were selected from each variant caller, discarded attributes from one caller could possibly outperform a "selected" attribute from another. Future work could address these drawbacks to further improve the model. Regardless, the F1 score of the current model demonstrates its reliability across the varied datasets.

References

1. Vijayan V, Yiu SM, Zhang L. Improving somatic variant identification through integration of genome and exome data. *BMC Genomics*. 2017;18(Suppl 7):748. Published 2017 Oct 16. doi:10.1186/s12864-017-4134-3
2. Wang, M., Luo, W., Jones, K. et al. SomaticCombiner: improving the performance of somatic variant calling based on evaluation tests and a consensus approach. *Sci Rep 10*, 12898 (2020). doi:10.1038/s41598-020-69772-8

Appendix

Table A1. *A total of 20 features were chosen to build the classifier, ranked based on Chi-square value. Features with prefix 'FILTER' were derived from output of variant callers. Features with prefix 'vs' come from VarScan, features with prefix 'fb' come from Freebayes, features with prefix 'vd' come from VarDict, features with prefix 'm2' come from Mutect2*

Chi-square	Attribute	Description
159545.9219	FILTER_Mutect2	classified as somatic by mutect2
151471.7605	FILTER_Vardict	classified as somatic by vardict
138576.5362	FILTER_Freebayes	classified as somatic by freebayes
119433.0628	vs_SPV	Fisher's Exact Test P-value of tumor versus normal for Somatic/LOH calls
93072.47285	vs_SSC	Somatic score in Phred scale (0-255) derived from somatic p-value
42162.73262	vd_SSF	P-value
30942.89578	f_DPRA	Alternate allele depth ratio. Ratio between depth in samples with each called alternate allele and those without.
28578.07305	m2_NLOD	normalised log odds
26366.72601	m2_MQ	**mapping quality** across all reads
25967.01141	vd_AF	Allele Frequency, for each ALT allele, in the same order as listed
23057.95678	m2_MQRankSum	Z-score From Wilcoxon rank sum test of Alt vs. Ref read mapping qualities
22592.92314	m2_DP	approx read depth at position
20443.66982	f_ODDS	The log odds ratio of the best genotype combination to the second-best
18821.42353	m2_HCNT	Number of haplotypes that support this variant
18108.56028	f_DP	Total read depth at the locus
18108.56028	f_DPB	Total read depth per bp at the locus; bases in reads overlapping / bases in haplotype
17624.74327	f_MQM	Mean **mapping quality** of observed alternate alleles
17090.54091	f_MQMR	Mean **mapping quality** of observed reference alleles
9066.4153	vd_MSI	Microsatellite
235.00614	FILTER_Varscan	classified as somatic by varscan

Table A2. *Precision, recall and F1 scores yielded from the naive approach on real1 and syn1-5 data. For F1 scores, the average was 0.54617, the highest was 0.8404 and the lowest was 0.2652.*

	real1	syn1	syn2	syn3	syn4	syn5
Precision	0.15495	0.34883	0.41575	0.37533	0.40153	0.74285
Recall	0.91963	0.98642	0.98776	0.94747	0.79344	0.96734
F1 score	0.26522	0.51539	0.58520	0.53767	0.53322	0.84036

Table A3. *Precision, recall and F1 scores yielded from J48 decision tree constructed on real1 and syn1-5 data, with leaf nodes containing at least 20 instances. For F1 scores, the average was 0.88306, the highest was 0.94966 and the lowest was 0.75252.*

	real1	syn1	syn2	syn3	syn4	syn5
Precision	0.80484	0.89958	0.82283	0.98242	0.95453	0.99981
Recall	0.70660	0.97258	0.97669	0.91902	0.70512	0.91821
F1 score	0.75252	0.93466	0.89318	0.94966	0.81108	0.95727

Table A4. *Precision, recall and F1 scores yielded from J48 decision tree constructed on real1 and syn1-5 data, with leaf nodes containing at least 20 instances, tested on real2_part1 without exclusion of any training data.*

	real2_part1
Precision	0.92639
Recall	0.74338
F1 score	0.82485

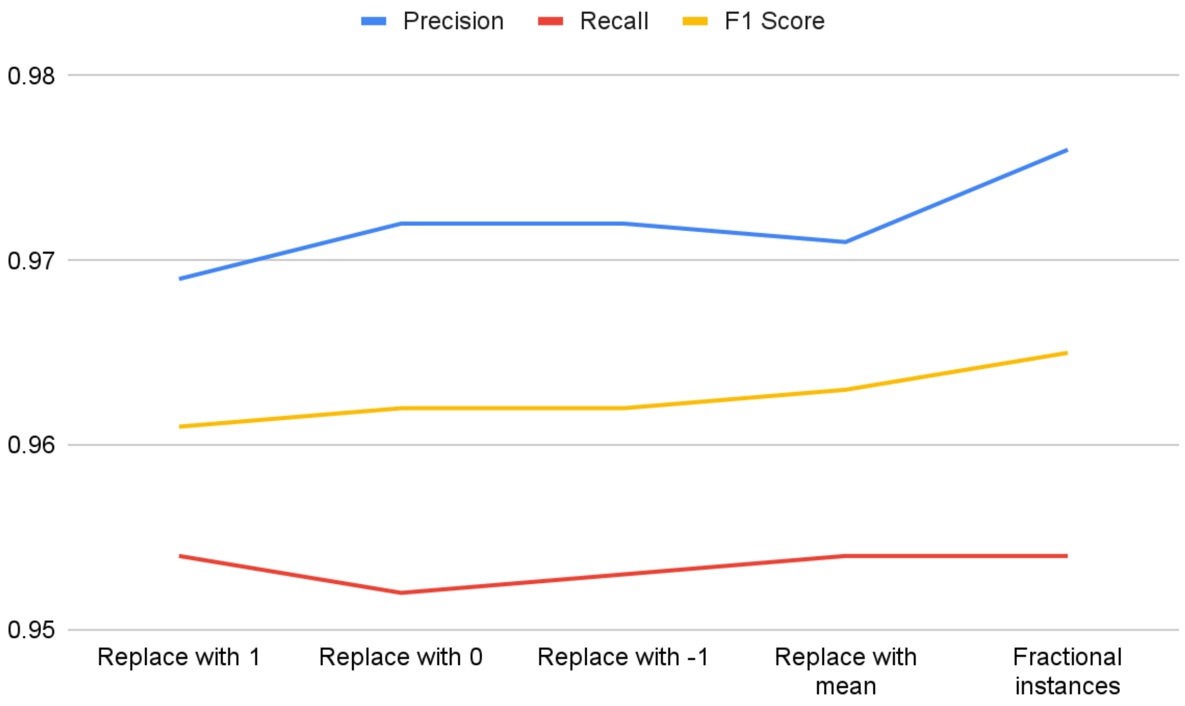

Figure A1. *Fractional Instances is found to be the most reliable means of dealing with missing data. Precision, recall and F1 score of the other methods were noticeably lower when replacing missing values with constant values 1,0, or -1, or the column mean, as compared to the fractional instances method*

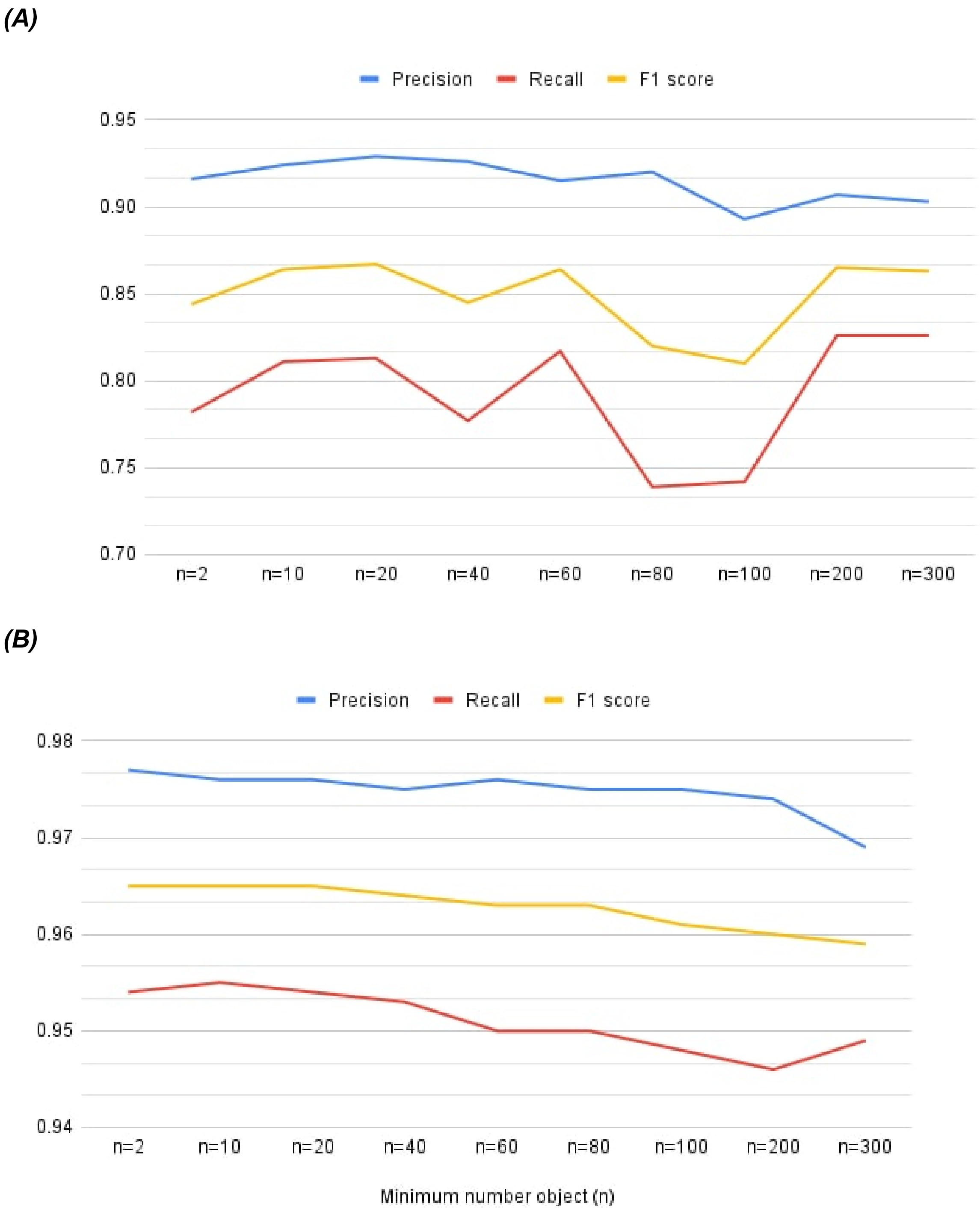

Figure A2. *Building J48 classifier with minimum number of objects of 20 gives the best F1 score. Precision, recall and F1 score of real2_part1 after pruning the decision tree. J48 trained with full dataset, (A) tested on real2_part1 (B) tested with 5-fold cross-validation*

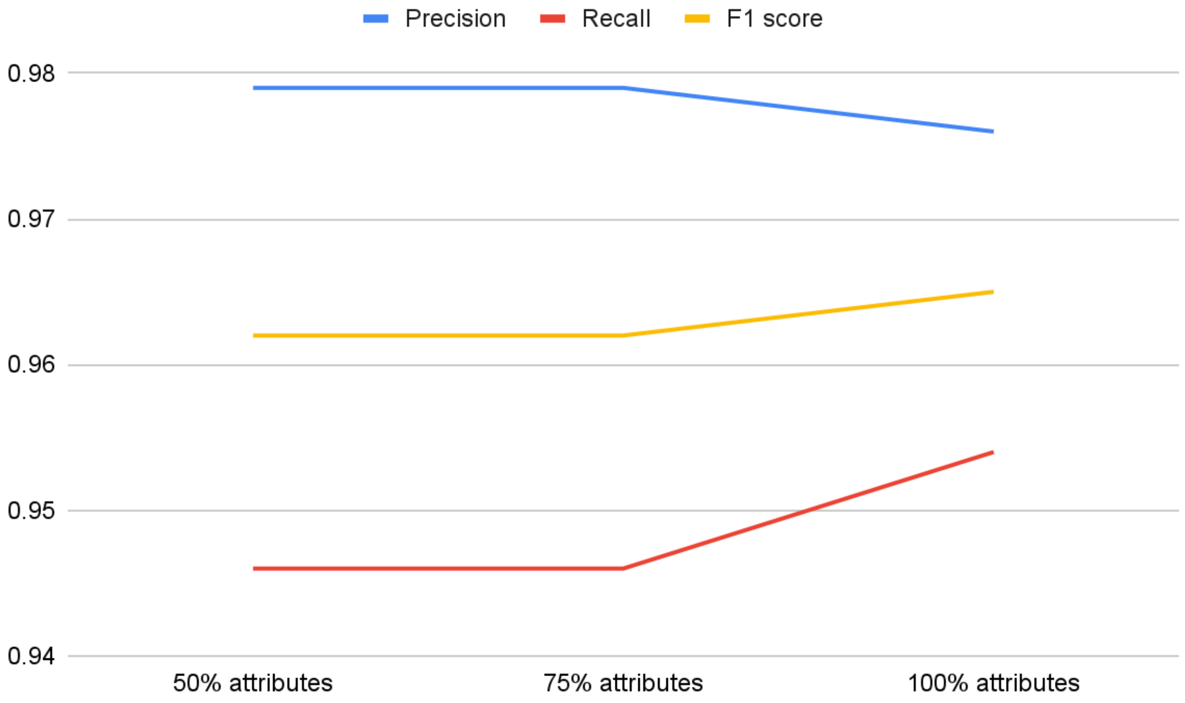

Figure A3 *Model trained with the full attribute list has the highest F1 score*