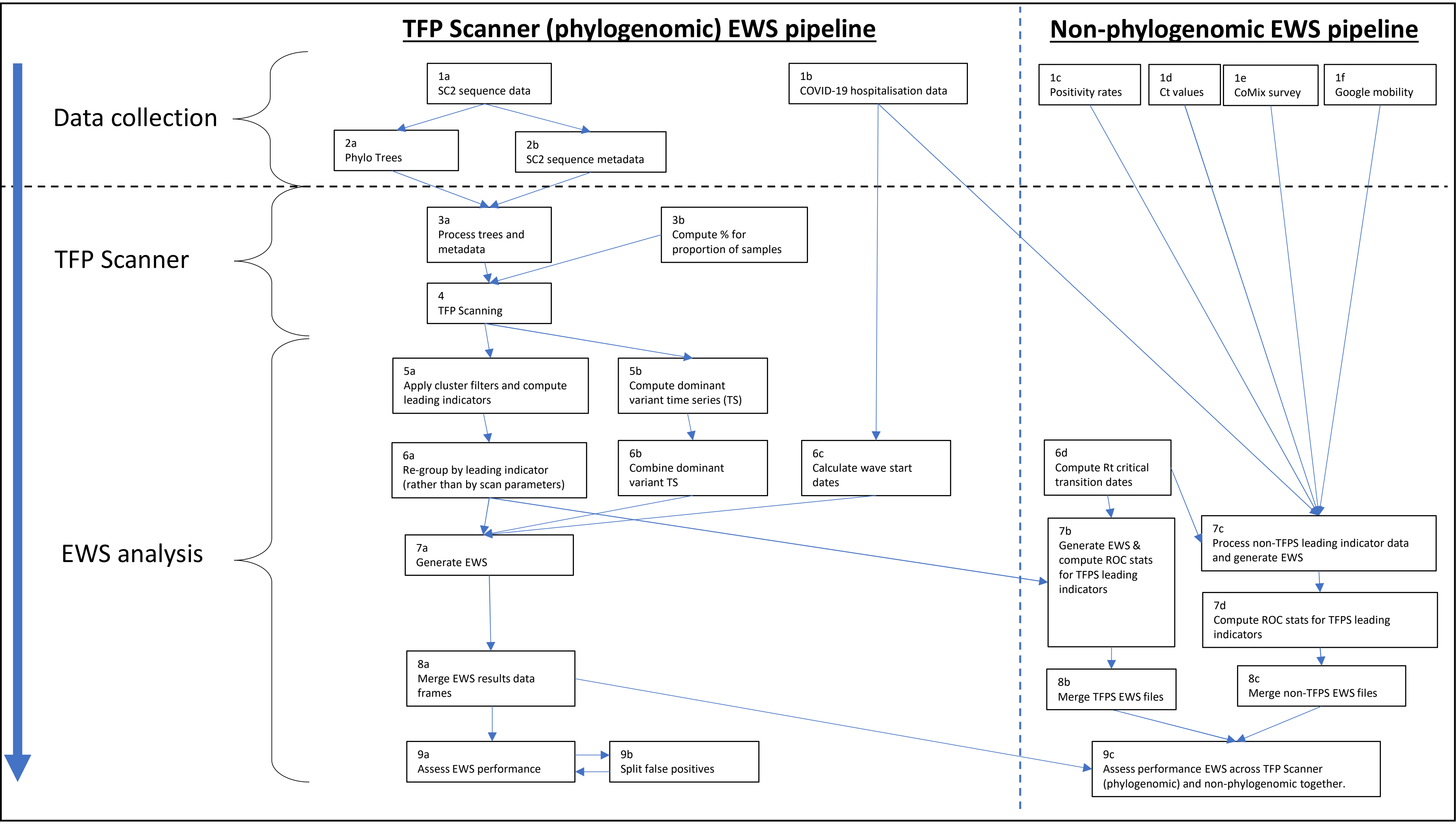
TFP Scanner (phylogenomic) EWS pipeline
Non-phylogenomic EWS pipeline
Data collection
TFP Scanner
EWS analysis
1a SC2 sequence data
1b COVID-19 hospitalisation data
1c Positivity rates
1d Ct values
1e CoMix survey
1f Google mobility
2a Phylo Trees
2b SC2 sequence metadata
3a Process trees and metadata
3b Compute % for proportion of samples
4 TFP Scanning
5a Apply cluster filters and compute leading indicators
5b Compute dominant variant time series (TS)
6a Re-group by leading indicator (rather than by scan parameters)
6b Combine dominant variant TS
6c Calculate wave start dates
6d Compute Rt critical transition dates
7a Generate EWS
7b Generate EWS & compute ROC stats for TFPS leading indicators
7c Process non-TFPS leading indicator data and generate EWS
7d Compute ROC stats for TFPS leading indicators
8a Merge EWS results data frames
8b Merge TFPS EWS files
8c Merge non-TFPS EWS files
9a Assess EWS performance
9b Split false positives
9c Assess performance EWS across TFP Scanner (phylogenomic) and non-phylogenomic together.

# TFP Scanner (phylogenomic) EWS Pipeline Flow Key

| Step | Name | Location | Description | Script | Input | Output |
|---|---|---|---|---|---|---|
| 1a | SC2 sequence data | | SARS-CoV-2 genomic sequence data from diagnostic testing<br>Source: COG-UK | | | |
| 1b | SC2 hospitalisation data | | SARS-CoV-2 UK hospitalisation data<br>Source: UK's Coronavirus dashboard (https://coronavirus.data.gov.uk/) | | data_2022-May-09 - UK Covid-19 hospital admissions.csv | |
| 2a | Phylo Trees | | Phylogenetic trees (288 trees between Aug 2020 and Mar 2022)<br>Source: CLIMB | | e.g. 'cog_global_2020-08-14_tree.newick' | |
| 2b | SC2 sequence metadata | | Genomic sequence metadata<br>Filename: cog_metadata_2022_05_03.csv.gz<br>Source: CLIMB | | | |
| 3a | Process trees and metadata | Laptop | Phylo tree and metadata processing.<br>Only UK Pillar 2 (P2) sequences and remove erroneous dates.<br>Includes calculation of most recent sample dates to be used as the max_date input for the tfpscan function. Based on 99.9% quantile of samples. | e.g. '1_tree_prep_pre_tfpscan_v5.R' | cog_metadata_2022_05_03.csv.gz<br>samples_to_be_removed_unique_name.rds | e.g. 'tfps-env-HPC-start_v5.Rdata' |
| 3b | Compute % for proportion of samples | Laptop | Calculate the percentages to be used to compute the proportion of samples such that the minimum value across trees for the minimum number of descendants = 20. | e.g. 'n_samples_by_date_v3.R' | | |
| 4 | TFP Scanning | HPC<br>(72hrs walltime but usually some get killed so need re-running) | Each phylogenetic tree input along with metadata into TFP Scanner multiple times with different TFP Scanner parameters | e.g. 'tfps_load_7_56_20.R' (a different script used for each different sets of parameters) and 'tfps_run_tre1_100_7_56_20.pbs' used to run array on high performance computing (HPC) resources with array input files, e.g. 'array_input_1_100.txt' and 'tre_list_1_100.txt'. Note that the 289 trees were split into 3 groups (1-100, 101-238 and 239-289). Use 'file_check.R' to check all files have been produced i.e. all trees have been scanned with all parameters. | 'tfps-env-HPC-start_v5.Rdata'<br>and<br>phylogenetic trees | e.g. 'scanner-2020-08-14-min_age_7-max_age_56-min_desc_20.rds' |
| 5a | Apply cluster filters and compute leading indicators | HPC<br>(c.3 hrs) | Each .rds file output (data frame) from the tfpscanner has a set of filters applied and leading indicator statistics are calculated. Done on HPC with a different R script for different filter parameters and array to cycle through all of the files with different scan parameters. | e.g.<br>'tfps_analysis_lgr_th_060_pval_filter_001_HPC_array.R'<br>With .pbs file for HPC array e.g. 'tfps_compile_time_series_large_adjust_060_pval_filter_001.pbs' and array input files e.g. 'array_input_1_24.txt' and 'folder_list.txt' | | data frames of stats for each tree and parameter/filter set |

# TFP Scanner (phylogenomic) EWS Pipeline Flow Key (2)

| Step | Name | Location | Description | Script | Input | Output |
|---|---|---|---|---|---|---|
| 5b | Compute dominant variant time series (TS) by parameter | HPC<br>(<3hrs) | Top variants. Create a database of the top 5 dominant SC2 variants by frequency of clusters contributing to the leading indicator signal (i.e. after cluster filtering) as well as top five dominant variants in clusters by logistic growth rate (positive only but including GAM and simple LGR). This is done over all TFPS parameters and filters. Results are used to determine whether an EWS is True or False positive. Ensure there are no 'environment' files in the folders when running on HPC. | e.g. 'top_variants_lgr_075_pval_001.R'<br>With .pbs file for HPC array e.g. top_variants_lgr_075_pval_001.pbs |  | e.g.<br>top_variants_mina07_maxa56_mind001_p_val_0.01_lgr_th_0.6.rds. |
| 6a | Re-group by leading indicator (rather than by scan parameters) | Laptop<br>(<1hr) | Manipulate leading indicator data from being grouped in a data frame by scan parameters to being grouped in data frames by leading indicator. | 'compile_dataframes_2023_02.R'<br>Note that the folder structure needs to be correct:<br>/large_cluster_adjust_lgr_threshold_xxx/p_val_filter_xxx/dataframes_statistics/ and /dataframes_variables/ and /lists_statistics/ and /lists_variables/ | e.g.<br>'tfps_min_age_7_max_age_56_min_desc_20_growth_var_df.rds'<br>etc. | e.g. 'tfps_lgr_gam_max.csv' etc. and<br>'tfps_growth_lists.rds' etc. |
| 6b | Combine dominant variant TS | Laptop<br>(c.10mins) | These are all then combined to make a single data frame | 'merge_top_variants_dataframes_2023_02.R' | e.g.<br>'top_variants_mina07_maxa56_mind001_p_val_0.01_lgr_th_0.6.rds' | 'top_variants.rds'<br>'top_variants_positive_df.rds' (includes only those with positive growth) |
| 6c | Compute wave start dates | Laptop | Compute COVID-19 epidemic wave start dates using new hospital admissions data.<br>'wave_define_overlay.R' is used as an overlay to input a range of values for variables so as to optimise the computation of the wave start dates. This script uses for functions to: load case or hospitalisation data ('data_load.R'), fit GAM ('gam_fitting.R'), and compute growth rates ('wave_define_growth_model.R'). | data_load.R<br>gam_fitting.R<br>wave_define_growth_model.R<br>wave define - overlay.R | COVID-19 hospitalisation data | Table of waves and dates which is manually entered into R scripts. |
| 7a | Generate EWS | HPC<br>(c.2hrs) | Measure leading indicator data (e.g. 'tfps_vlgr_samp.csv') against range of EWS threshold levels and generate EWS signal. Determine whether true or false positive EWS by comparing against 'top_variants.rds'. | e.g.<br>'EWS_calc_threshold_lgr_th_060_pval_filter__001_HPC_array.R'<br>With .pbs file for HPC array e.g.<br>'tfps_EWS_lgr_th_060_pval_filter_001.pbs'<br>And leading indicator data e.g.<br>'tfps_vlgr_samp.csv' | 'top_variants.rds'<br>'top_variants_positive_df.rds' (includes only those with positive growth) | 'ews_results…rds'.<br>'wave_results_analysis…rds' and 'wave_results…rds' files for each leading indicator and lgr_th and p_val parameter set. |
| 8 | Merge EWS results dataframes | Laptop<br>(<30mins) | Merge the 1710 .rds files produced in step 7a. Three data frames (ews_results…rds, wave_results_analysis…rds and wave_results…rds) for 19 leading indicators, 10 LGR thresholds  and 3 p-value thresholds. | merge_results_dataframes_2023_03.R | 1710 .rds files of format<br>'ews_results…rds'.<br>'wave_results_analysis…rds' and<br>'wave_results…rds' | 'ews_results_df.rds'<br>'wave_results_analysis_reshaped.rds'<br>'wave_results_df.rds'<br>'wave_results_analysis_df.rds' |

# TFP Scanner (phylogenomic) EWS Pipeline Flow Key (3)

| Step | Name | Location | Description | Script | Input | Output |
|---|---|---|---|---|---|---|
| 9a | Assess EWS performance | Laptop | Extract information from EWS results data frames by ranking and filtering and then outputting tables for inspection. | wave_results_analysis_Mar_2023.R | ‘wave_results_analysis_reshaped.rds’<br>‘wave_results_analysis_df.rds’ | EWS false positive vs earliest true positive EWS plots<br>Tables of leading indicators ranked by lead time and false positives |
| 9b | False positive split | Laptop -> | Calculate the number of false positives that are before and after the earliest true positive for each leading indicator parameter set. Environment prepared on laptop. | 1_false_positive_split_prep_for_HPC.R | ‘wave_results_analysis_reshaped.rds’<br>‘wave_results_analysis_df.rds’ | false_positive_split_env_HPC_start_v2.Rdata |
| | | HPC -> | Computation of values on HPC. | 2_false_positive_split_function_HPC_v2.R | false_positive_split_env_HPC_start_v2.Rdata | .rds of format "output_df_w",wave_n,"_",row_start,"_",row_end,".rds“ |
| | | Laptop | Merger of resulting data frames on laptop. | 3_false_positive_split_df_merge.R | .rds of format "output_df_w",wave_n,"_",row_start,"_",row_end,".rds“ | |

# Non-phylogenomic EWS Pipeline Flow Key

| Step | Name | Location | Description | Script | Input | Output |
|---|---|---|---|---|---|---|
| 1c | Positivity rate | Laptop | Prepare positivity rate data for analysis | Positivity_rate_calc.R | | Sc2_positivity_rate_England.rds |
| 1d | Ct values | Laptop | Process the Ct value data to create potential leading indicator time series | Ct_values_processing.R | | Ct_p2_median_df_v2.csv<br>Ct_p2_mean_df_v2.csv |
| 1e | CoMix survey | | Summary analysis of CoMix survey data provided by Christopher Jarvis of the London School of Hygiene & Tropical Medicine | | 2022-03-02_bs_means_2w_open.csv | |
| 1f | Google mobility | Laptop | Prepare Google mobility data for analysis. Simple merge of calendar year files. | | 2020_GB_Region_Mobility_Report.csv<br>2021_GB_Region_Mobility_Report.csv<br>2022_GB_Region_Mobility_Report.csv | Google mobility data - UK to 12 June 2022.csv |
| 6d | Compute Rt critical transition dates | Laptop | Compute Rt critical transition dates (the dates at which Rt moves from <1 to >1. Weekly Rt data sourced from the UK Coronavirus dashboard (https://coronavirus.data.gov.uk/) | Rt_fill_days.R | 221207_R_and_growth_rate_time_series_for_publication_v1.0.ods<br>Rt.csv (containing 3 columns: date, lower, upper [bounds for UK values for Rt] | Rt Critical transition dates can be read from output time series |
| 7b | Generate TFPS time window EWS | HPC | Generate ROC stats for EWS generated using TFPS leading indicators. Uses .pbs files to perform analysis on HPC using an array of jobs so that multiple runs and parameter values can be run at the same time. | EWS_gen_time_window_TFPS_multi_stat_HPC.R<br>EWS_gen_time_window_TFPS_multi_stat_HPC_10d.R<br>roc_multi_stat.pbs<br>roc_multi_stat_10d.pbs | TFPS leading indicator .csv files, e.g. "tfps_vlgr_samp.csv" or "tfps_vlgr_simple_samp.csv" | "ROC_multi_stat_sd_df_lgr_th_",pslt,"_pval_th_",pvt,".rds“<br>"ROC_multi_stat_rt_df_lgr_th_",pslt,"_pval_th_",pvt,".rds" |
| 7c | Process non-TFPS leading indicator data and generate EWS | Laptop | Prepare non-TFPS leading indicator data and use to generate early warning signals (EWS) and assess (TP,TN,FP,FN) using a time window.<br>#' 1 - Load data for each leading indicator type and format in dataframes with date and different variations in columns<br>#' 2 - Compute 'robust' z-scores for all leading indicator data<br>#' 3 - Compute whether z-score above or below EWS threshold<br>#' 4 - Load waves definitions: start/inflection date, Rt critical transition dates, wave band dates<br>#' 5 - For each leading indicator, wave, and EWS threshold, want to know number of TP, FP, TN and FN as well as earliest TP. | 1_EWS_gen_time_window_non_TFPS_processing.R | data_2022-May-09 - UK Covid-19 hospital admissions.csv<br>Ct_p2_mean_df_v2.csv<br>Ct_p2_median_df_v2.csv<br>sc2_positivity_rate_England.rds<br>2022-03-02_bs_means_2w_open.csv<br>Google mobility data - UK to 12 June 2022.csv | li_z_ews_sd_classification_array<br>li_z_ews_rt_classification_array<br>date_index_total<br>date_index_total_ex_B.1.177<br>date_index_total_ex_B.1.177_BA.1 |

# Non-phylogenomic EWS Pipeline Flow Key (2)

| Step | Name | Location | Description | Script | Input | Output |
|---|---|---|---|---|---|---|
| 7d | Compute ROC stats for non-TFPS | Laptop | Compute receiver operating characteristic (ROC) stats.<br>1 - Loop through the two different ways of calculating the time window (anchoring to Rt critical transition date or hospitalisation wave inflection date)<br>2 - Loop through date ranges (individual waves, inc all waves, all waves exc. B.1.177 and all waves exc B.1.177 & BA.2)<br>3 - Loop through leading indicators<br>4 - Calculate ROC stats and AUC for each wave etc. | 2_EWS_gen_time_window_non_TFPS_multi_stat.R | li_z_ews_sd_classification_array<br>li_z_ews_rt_classification_array<br>date_index_total<br>date_index_total_ex_B.1.177<br>date_index_total_ex_B.1.177_BA.1 | ROC_stats_sd_non_tfps_5d.rds<br>ROC_stats_rt_non_tfps_5d.rds<br>ROC_stats_sd_non_tfps_10d.rds<br>ROC_stats_rt_non_tfps_10d.rds |
| 8b | Merge TFPS EWS ROC stats | Laptop | Merges the data frames containing the ROC stats calculated for the TFPS leading indicators. | merge_ROC_multi_stat_HPC_outputs.R | | ROC_multi_stat_sd_tfps_combined_5d.rds<br>ROC_multi_stat_rt_tfps_combined_5d.rds<br>ROC_multi_stat_sd_tfps_combined_10d.rds<br>ROC_multi_stat_rt_tfps_combined_10d.rds |
| 8c | Merge non-TFPS EWS ROC stats | Laptop | Merges the data frames containing the ROC stats calculated for the non-TFPS leading indicators. | merge_results_dataframes_non_TFPS_2023_03.R | | ROC_stats_sd_non_tfps_5d.rds<br>ROC_stats_sd_non_tfps_10d.rds<br>ROC_stats_rt_non_tfps_5d.rds<br>ROC_stats_rt_non_tfps_10d.rds |
| 9c | Merge TFPS and non-TFPS ROC stats and rank | Laptop | Merge multiple ROC stats for genomic and non-genomic leading indicators calculated using different time windows (wave start date and Rt critical transition anchor points, and t+5 and t+10 days). Rank the leading indicator parameter sets using normalised MCC. | 3_ranking_leading_indicators.R | ROC_multi_stat_sd_tfps_combined_5d.rds<br>ROC_multi_stat_rt_tfps_combined_5d.rds<br>ROC_multi_stat_sd_tfps_combined_10d.rds<br>ROC_multi_stat_rt_tfps_combined_10d.rds<br>ROC_stats_sd_non_tfps_5d.rds<br>ROC_stats_sd_non_tfps_10d.rds<br>ROC_stats_rt_non_tfps_5d.rds<br>ROC_stats_rt_non_tfps_10d.rds | combined_df.rds<br>combined_df_order_perc_rank_w3_4_5_6_7_top1000.csv<br>combined_df_order_perc_rank_w3_4_7_top1000.csv<br>combined_df_order_perc_rank_w5_6_top1000.csv<br>rank1_genomic_ranking.csv<br>rank2_genomic_ranking.csv |