Stability Analysis Vitalizing Instability Classification - `SAVIC` - software
Version 1.00
February 26, 2023

This software is available on
`https://github.com/MihailoMartinovic/SAVIC`

1 Background & Motivation

The understanding of the solar wind plasma kinetics is based on examining the Velocity Distribution Function (VDF) of particles. If the state of VDF is sufficiently different from Local Thermodynamic Equilibrium (LTE), then the particle energy can be emitted through through fields—waves are excited and particles lose energy, thereby getting "pushed" toward LTE [Stix, 1992]. Here, we label the energy that is contained in particles that can be emitted through unstable wave modes "free energy". VDF characteristics, such as temperature anisotropy, particle drifts, or excess pressure in a given direction can act as sources of free energy.

Traditionally, prediction of unstable modes is performed by dispersion solvers, who aim to map the dispersion relation of the VDF dispersion relation—$\underline{\mathbf{D}}(\omega, \mathbf{k})$, where $\omega = \omega_{\mathrm{r}} + i\gamma$ is complex frequency and $\mathbf{k}$ is wavevector—over the ω and $\mathbf{k}$ phase space. If at any point in this space we find $\gamma > 0$, the mode is unstable. The calculations of traditional solvers [Roennmark, 1982, Quataert, 1998, Klein and Howes, 2015, Verscharen and Chandran, 2018] can be done up to an arbitrary level of precision, but generally require a large volume of CPU hours, making them less convenient for analysis of larger data sets. Also, such codes are not user friendly, and require a notable time investment before a user becomes familiar. A practical consequence of these facts is that usage of the dispersion solvers—even though they are mostly publicly available—is limited almost exclusively to authors of these codes and their respective research groups.

To overcome these difficulties, `SAVIC` software accomplishes three major objectives:

- use Machine Learning (ML) algorithms to decrease the requirements for the computational power required by traditional solvers by several orders of magnitude, providing the instability properties for a given VDF practically instantaneously

- provide the user friendly environment that would make access to plasma stability analysis tools easy for the entire community
- make additional step in the analysis of the numerical results provided by dispersion solvers by classifying the type of the predicted unstable mode

2 Outline of the project

The `SAVIC` software intends to combine:

1. Predictor of the VDF stability — `SAVIC-P`
2. Regressors that quantify the parameters of the MUM — `SAVIC-Q`
3. Classifiers that recognize the type of instability predicted for the given VDF — `SAVIC-C`

`SAVIC` software is a combination of the three components listed above. However, each of the given codes can also be used separately.

3 Input files and variables; loading the data

`SAVIC` recognizes input files that are loaded in `Python Pandas` DataFrame with the set of variables (column names) as follows

core	var name	`beta_par_core`	`alph_c`		
	quantity	$\beta_{\parallel,c}$	$\frac{T_{\perp,c}}{T_{\parallel,c}}$		
beam	var name	`tau_b`	`alph_b`	`D_b`	`vv_b`
	quantity	$\frac{T_{\parallel,c}}{T_{\parallel,b}}$	$\frac{T_{\perp,b}}{T_{\parallel,b}}$	$\frac{n_b}{n_c}$	$\frac{\Delta v_{b,c}}{v_{Ac}}$
α	var name	`tau_a`	`alph_a`	`D_a`	`vv_a`
	quantity	$\frac{T_{\parallel,c}}{T_{\parallel,\alpha}}$	$\frac{T_{\perp,\alpha}}{T_{\parallel,\alpha}}$	$\frac{n_\alpha}{n_c}$	$\frac{\Delta v_{\alpha,c}}{v_{Ac}}$

Table 1: Input variables for `SAVIC` codes.

Any combination of the four types of input data frames (C, CB, Cα, CBα) can be loaded at the same time (see `00_SAVIC_Input_Sort.ipynb` example), but the ordering of the columns must be as shown in Table 1.

The input is loaded by recognizing the set of variables, and henceforth launching one if four available versions of `SAVIC` —core only (C), core and beam (CB), core and α (Cα, core, beam, and α (CBα)). This task is handled by the *SAVIC_Input_Sort* function inside `000_SAVIC.ipynb` (further on referred to as the main code), and illustrated in `00_SAVIC_Input_Sort.ipynb`.

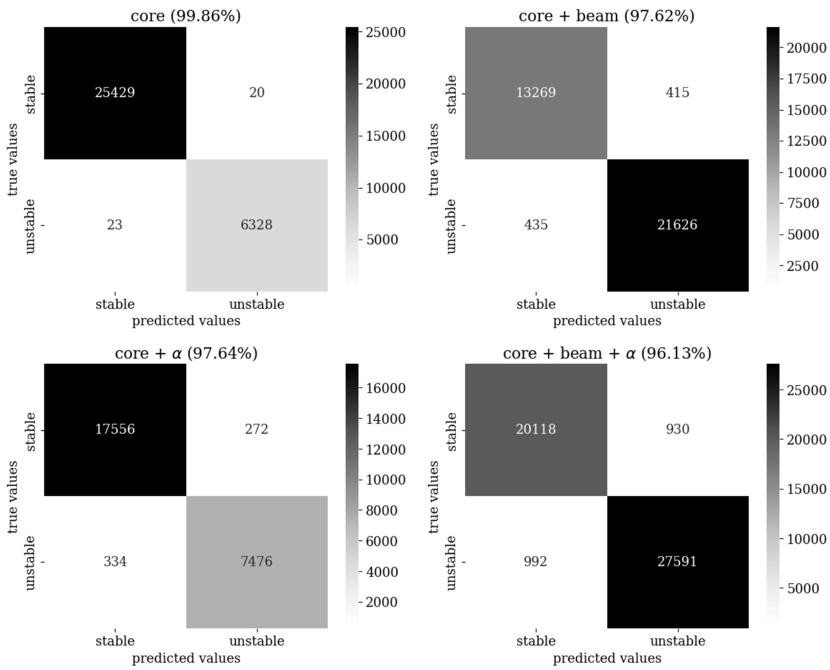

Figure 1: Confusion Matrices of plasma stability predicted by `SAVIC-P` for the four ion VDF subsets, compared against `PLUMAGE`-derived instability calculations as the "true" values.

4 How it works?

4.1 Stability prediction - `SAVIC-P`

`SAVIC-P` is an instability predictor that aims to distinguish if a given VDF is either stable or unstable. It was trained and tested using `PLUMAGE` software [Klein et al., 2017] applied to *Helios* fitted VDF database [Ďurovcová et al., 2019]. Confusion Matrices (CM) for all four versions of the code are given in Figure 1.

The output of `SAVIC-P` is an additional column in the input matrix, labeled `'unstable'`, with *True/False* values. Examples needed for running of this code are given in `00_SAVIC-P.ipynb`, and the step-by-step breakdown of the code in `00_SAVIC-P-tutorial.ipynb`.

4.2 Quantifying unstable mode parameters - `SAVIC-Q`

4.2.1 `SAVIC-Q` Preclassifiers

`SAVIC-Q` assumes that the input VDFs are unstable, and will proceed to quantify the instability parameters. In the code development, the main limit in `SAVIC-Q` accuracy was having both positive and negative values of P_C, P_B, and P_α

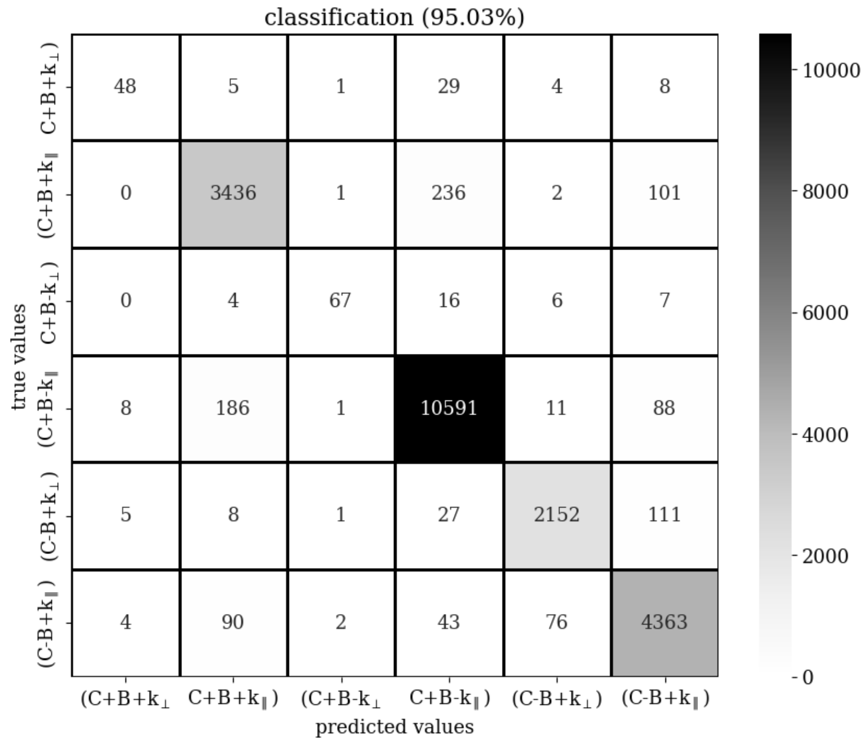

Figure 2: Classification of unstable modes for the CB data subset. The "+" sign for core and beam signals that the component is emitting power, while $k_\perp$ and $k_\parallel$ stand for oblique and parallel propagation, respectively. The groups that share the same SAVIC-Q regressor are marked by brackets.

(quantities explained in the paper). To overcome this issue, a new set of classifiers is built with sole objective to distinguish which VDF component (C, B, α, or any combination of the three) emits energy, and if the unstable mode is either parallel or oblique. Figures 2, 3, and 4 are display the test results of SAVIC-Q classifiers. The "+" sign signals that the component is emitting power, while $k_\perp$ and $k_\parallel$ stand for oblique and parallel propagation, respectively. In general, every category (preclassifier CM field) shown on this set of Figures should have its own SAVIC-Q trained regressor. However, some of the categories are grouped due to insufficient amount of training data. The groups of categories are shown in brackets on the axis of CMs. Once preclassified, the data is passed over to an adequate regressor.

4.2.2 SAVIC-Q Regressors

Each SAVIC-Q regressors uses the output of SAVIC-P as an input, and proceeds to estimate the logarithms of:

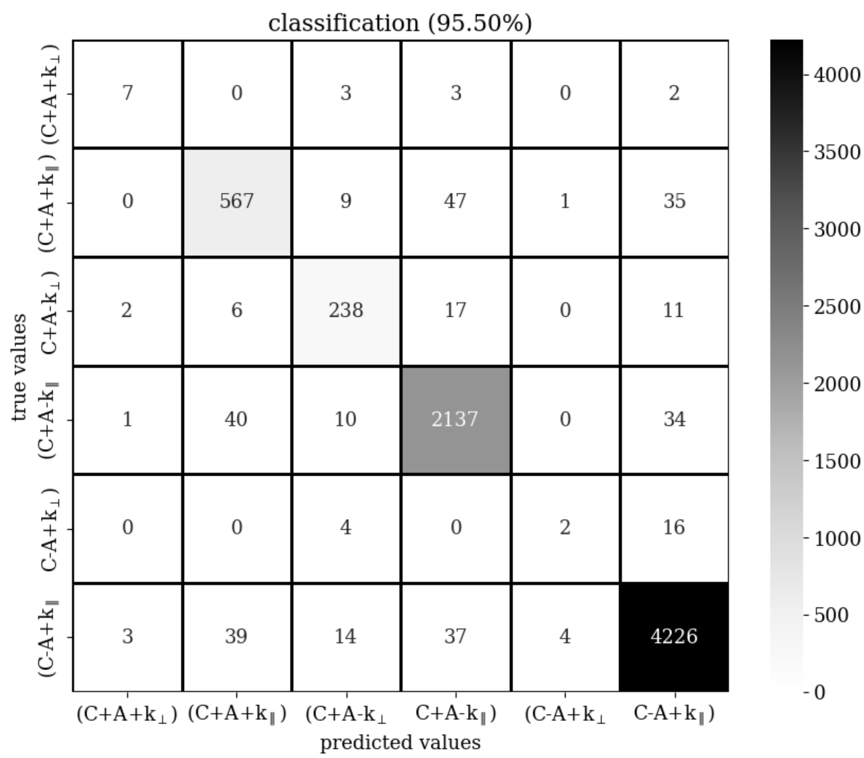

Figure 3: Classification of unstable modes for the Cα data subset. The labelling scheme is the same as on Figure 2.

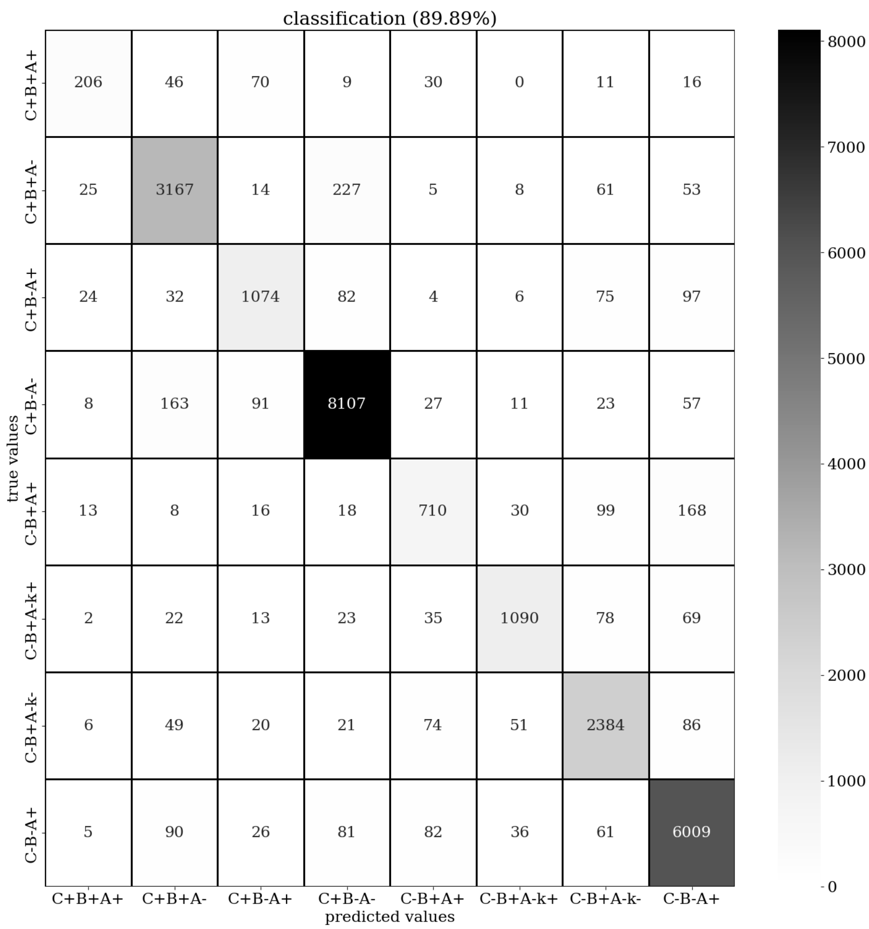

Figure 4: Classification of unstable modes for the CBα data subset. Each of the groups has its own corresponding trained regressor.

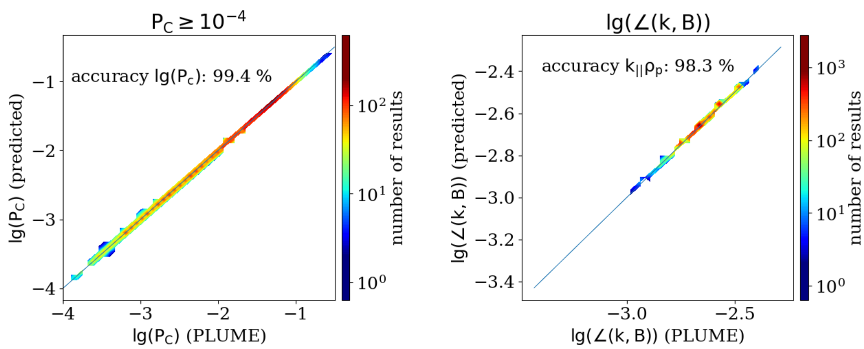

Figure 5: SAVIC-Q C regressor.

- the emitted power of the component that is expected to emit energy (P_C, P_B, P_α, or any combination of the three)
- the angle between the wavevector and the magnetic field for MUM $\theta_{kB}^{\max}$

The sets of SAVIC-Q regressors, along with their respective confidence levels, are shown on Figures 5-9. The amount of data used to train each regressor is roughly 9 times the number given in its corresponding field (or sum of fields) on the preclassifier CMs. The results of SAVIC-Q are then used as an input to SAVIC-C clustering algorithms. Examples needed for running of this code are given in 00_SAVIC-Q.ipynb, and the step-by-step breakdown of the code in 00_SAVIC-Q-tutorial.ipynb.

4.3 Identification of the unstable mode types - SAVIC-C

The Gaussian Mixture clustering algorithms are informed by the VDF parameters and SAVIC-Q output. The four subsets—C, CB, Cα, and CBα—have 4, 8, 6, and 12 clusters created to represent physically meaningful grouping of the the intervals by different types of instabilities (or groups of instabilities). The breakdown of various instability types is given throughout the literature (see e.g. [Gary, 1993, Klein, 2013, Verscharen et al., 2019] for useful reviews).

The outputs are labeled on Figure 10, and the adequate labels, which are in fact short textual descriptions of each cluster), are provided in SAVIC-C output data frames.

Here, it is important to emphasize that the uncertainty of clustering is not possible to estimate accurately. Therefore, the user that intends to apply SAVIC-C on a limited number of cases should use the clustering output as a first proxy before examining the results "by eye", and the user that processes large number of VDFs should keep in mind that SAVIC-C will provide statistically meaningful insight, but there is no guarantee that every individual interval is classified correctly. Examples needed for running of this code are given in 00_SAVIC-C.ipynb, and the step-by-step breakdown of the code in 00_SAVIC-C-tutorial.ipynb.

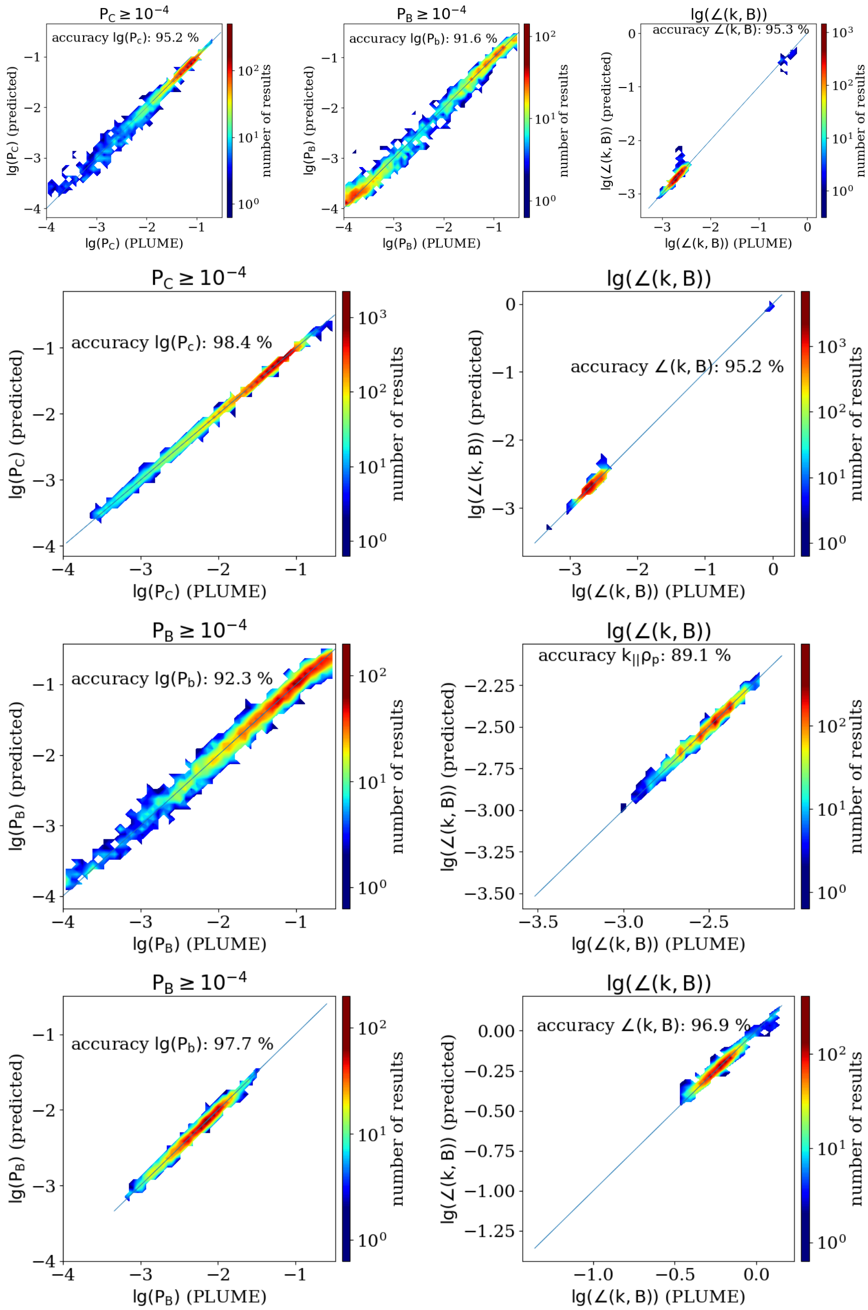

Figure 6: `SAVIC-Q` CB regressors: "C+B+", "C+B-", "C-B+$k_\parallel$", and "C-B+$k_\perp$", respectively.

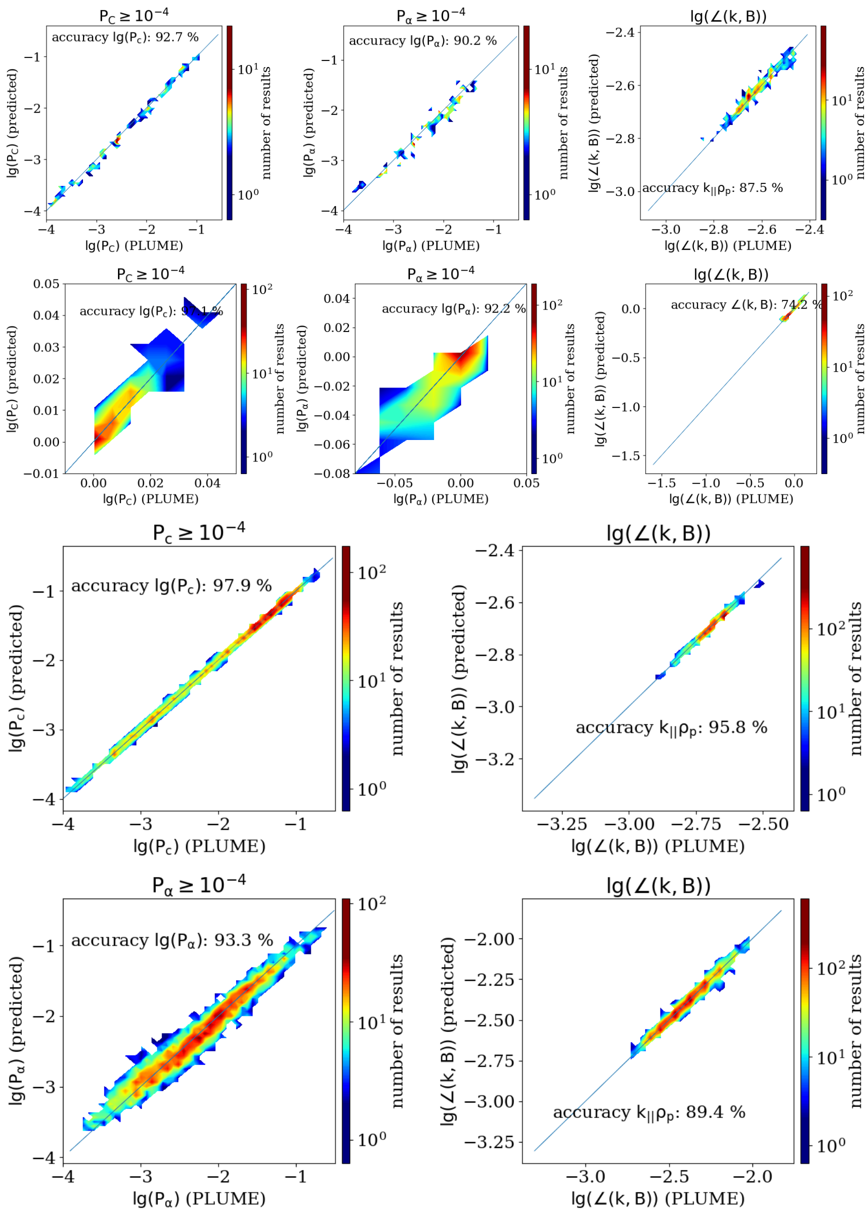

Figure 7: `SAVIC-Q` Cα regressors: "C+α+$k_{\parallel}$", "C+α+$k_{\perp}$", "C+α-", and "C-α+", respectively.

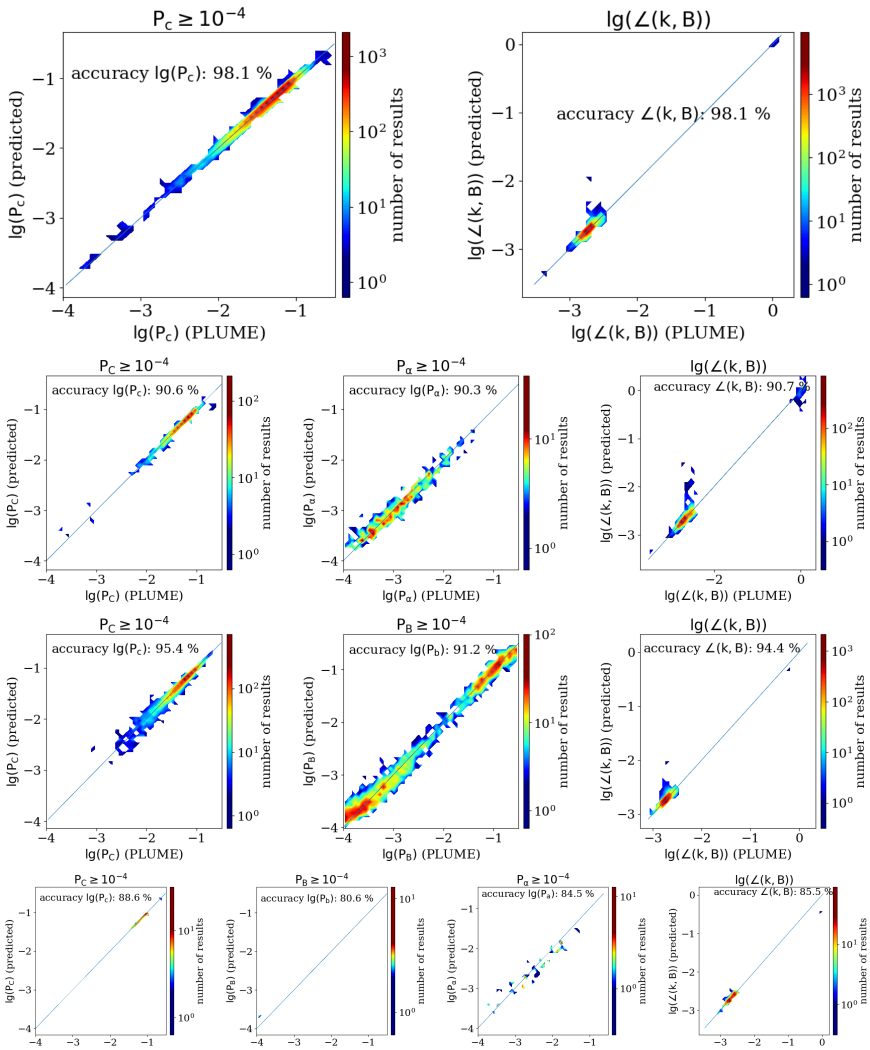

Figure 8: `SAVIC-Q` CBα C+ regressors: "C+B-α-", "C+B-α+", "C+B+α-", and "C+B+α+", respectively.

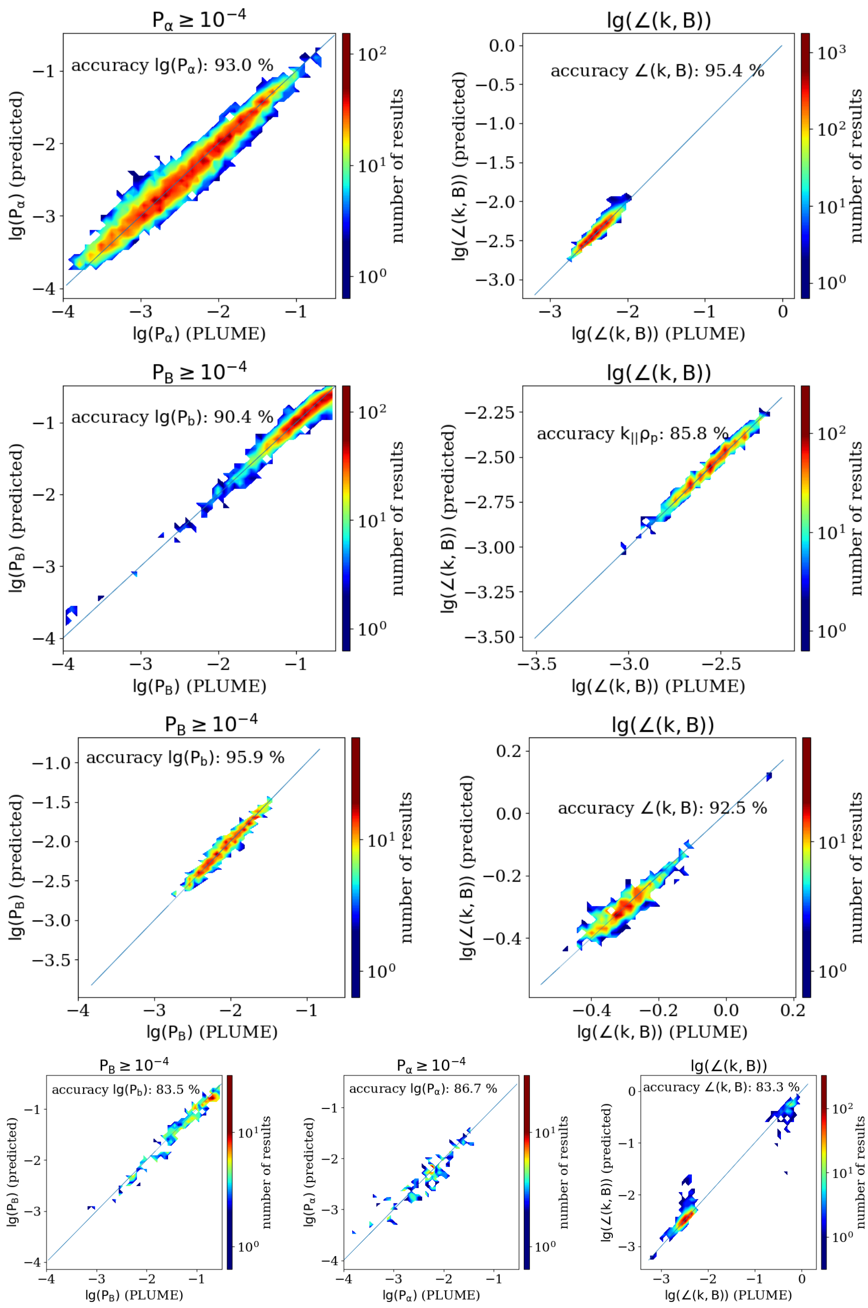

Figure 9: `SAVIC-Q` CBα C- regressors: "C-B-α+", "C-B+α-", "C-B+α-", and "C-B+α+", respectively.

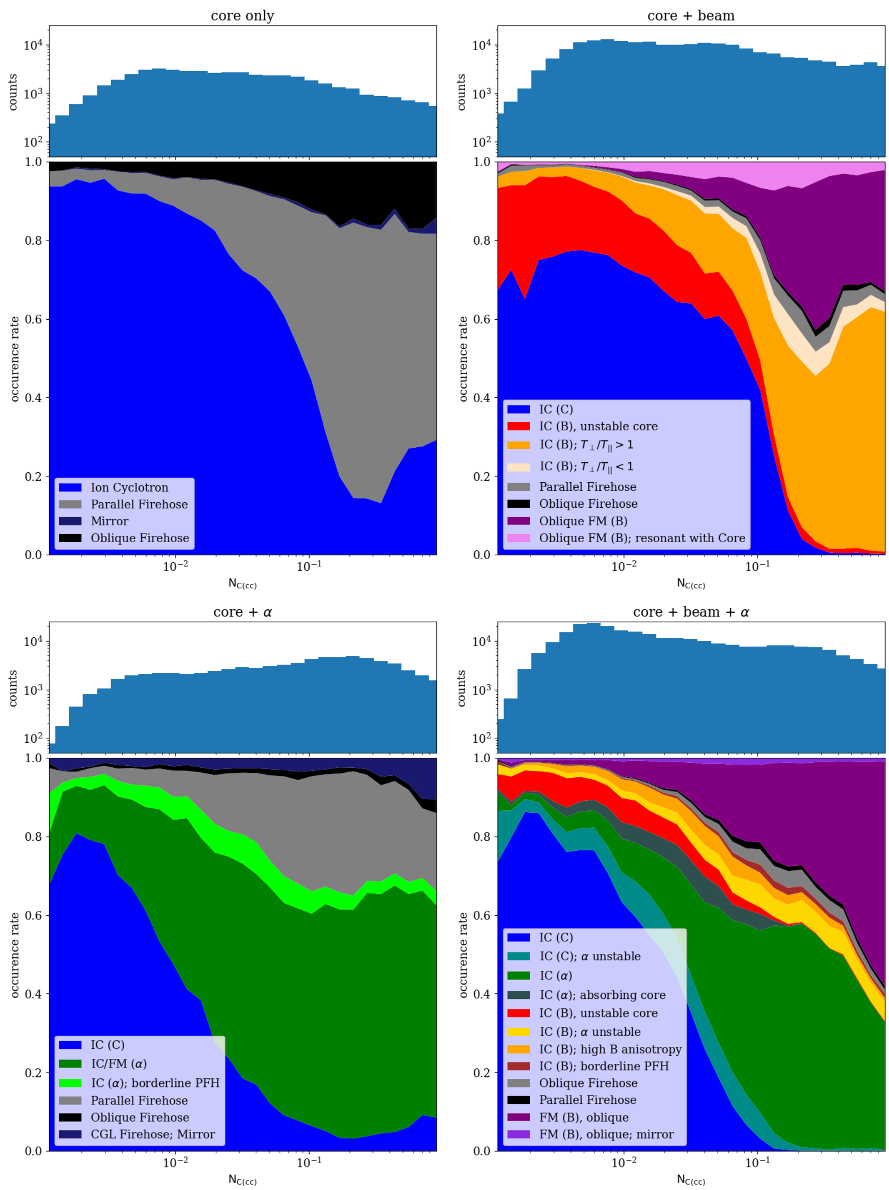

Figure 10: Overview of the instability clusters for all four subsets identified by `SAVIC-C`. The lack of α induced oblique modes is due to VDF fitting methods, which conflates thermal and beam populations of α particles into a single Maxwellian.

5 SAVIC Repository Files

This Section contains full descriptions of SAVIC repository files.

- Tutorials: every step of the code is given in a separate notebook cell and all the intermediate steps can be displayed.
  - 00_SAVIC-P-tutorial.ipynb
  - 00_SAVIC-Q-tutorial.ipynb
  - 00_SAVIC-C-tutorial.ipynb
- Partial code files: loaded models and running functions. Intended for sepatate use by advanced users.
  - 00_SAVIC-P.ipynb
  - 00_SAVIC-Q.ipynb
  - 00_SAVIC-C.ipynb
- Subset files: integrated SAVIC code for each subset of the data. Use if you already have input files adequate for a single subset.
  - 00_SAVIC_Core.ipynb
  - 00_SAVIC_CoreBeam.ipynb
  - 00_SAVIC_CoreAlpha.ipynb
  - 00_SAVIC_CoreBeamAlpha.ipynb
- The main code: fully integrated SAVIC code. Intended for basic users and quick use. Runs a full chain of algorithms as described in Section 4.
  - 000_SAVIC.ipynb
- Support files:
  - SAVIC_Python_Startup.py: run as a first cell in each notebook. Loads prerequisites.
  - 00_SAVIC_Input_Sort.py: regulates the input data frame.
  - 00_SAVIC_Examples/SAVIC_Examples.h5: Contains input data frame for each of the codes. Can be re-created in tutorial notebooks.

6 More to Come...

The models explained here will be updated as their shortcomings are being found, both via our testing and feedback from the community.

Additional features intended to be added are:

- mapping of confidence levels vs location in the phase space
- detailed explanation of uncertainty calculation (based on previous point)
- report on the upgrades from training on *Wind* data set

References

[Gary, 1993] Gary, S. P. (1993). *Theory of Space Plasma Microinstabilities*. Cambridge University Press.

[Klein, 2013] Klein, K. G. (2013). *The kinetic plasma physics of solar wind turbulence*. PhD thesis, The University of Iowa.

[Klein and Howes, 2015] Klein, K. G. and Howes, G. G. (2015). Predicted impacts of proton temperature anisotropy on solar wind turbulence. *Physics of Plasmas*, 22(3):032903.

[Klein et al., 2017] Klein, K. G., Kasper, J. C., Korreck, K. E., and Stevens, M. L. (2017). Applying Nyquist's method for stability determination to solar wind observations. *Journal of Geophysical Research (Space Physics)*, 122(10):9815–9823.

[Quataert, 1998] Quataert, E. (1998). Particle Heating by Alfvénic Turbulence in Hot Accretion Flows. *The Astrophysical Journal*, 500:978–991.

[Roennmark, 1982] Roennmark, K. (1982). Waves in homogeneous, anisotropic multicomponent plasmas (WHAMP). Technical Report.

[Stix, 1992] Stix, T. H. (1992). *Waves in plasmas*. Springer.

[Ďurovcová et al., 2019] Ďurovcová, T., Šafránková, J., and Němeček, Z. (2019). Evolution of Relative Drifts in the Expanding Solar Wind: Helios Observations. *Solar Physics*, 294(7):97.

[Verscharen and Chandran, 2018] Verscharen, D. and Chandran, B. D. G. (2018). NHDS: The New Hampshire Dispersion Relation Solver. *Research Notes of the American Astronomical Society*, 2(2):13.

[Verscharen et al., 2019] Verscharen, D., Klein, K. G., and Maruca, B. A. (2019). The multi-scale nature of the solar wind. *Living Reviews in Solar Physics*, 16(1):5.