MorphSeq Param Sweep with Splines;Dec-17-2024

`javascript`

---

type: daily
date: 2024-12-17
project: #morphSeq
date: Dec-17-2024

Dec 17 2024

MorphSeq Param Sweep with Splines

Previous Paramater Sweep (where we left off)

This is following up on the previous paramater sweep. Since then I have generated splines and different ways to measure colinearity WITHIN and across the splines for perturbations.
Main takeaways from previous param sweep:

> ### Main Takeawats
>
> - Classification is not a trivial task and there is a spread in model performance
> - We can have "pretty-morphology spaces that are good at classification
> - Right now we dont have a very good way of measuring "pretty"
> - NT-Xent this was the top performing model.
> - We dont have a good way to measure "prettyness" nor different type of it
>
> - There is some signal because Clustering coeffient (measure of cliquieness) did have a trend potentially.
> - Are there model archoctural deicsion that owuld impact classification reproducibility
>
> Questions:
>
> - why do all the "prettiest" best models have the highest value for Beta

> - I eventually how well deep learning vs log reg for classification
> - Does Aesthetic of the model matter
> - Stability
> - Should we measure these dim reduction space versus

Main Takeaway

Key Players

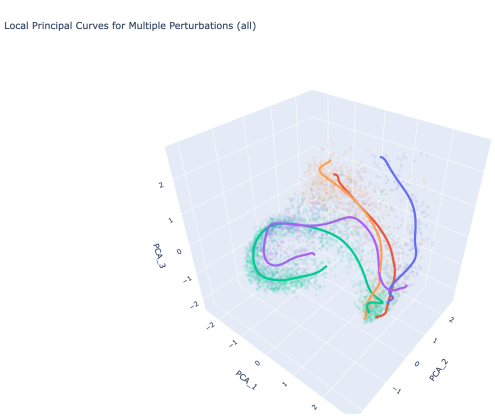
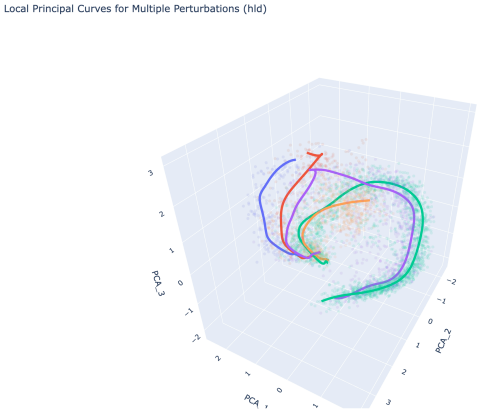
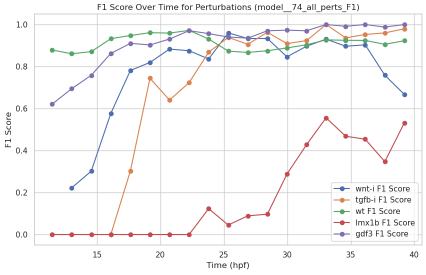
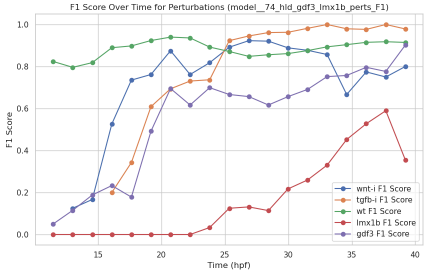
I used splines using @einbeckLocalPrincipalCurves20E05 to fit splines to the data we wanted to Measure

- Tree Likeness (Prettiness), done by looking at dispersion (distance to centroid) of splines
  - dispersion_first_n and dispersion_last_n is how close the points are in the first n points along the datasets splines (or lastn)
    - Note i used n=5
  - disp_coefficient is a fitting a linear model to the dispersion ALONG the spline
    - This is to make sure that splines (and therefore points) are spreading out over timne
  - SegmentColiniartity_mean_within (all or hld): point moving in same direction
    - This is looking WITHIN a dataset and looking at pairwise between perts how these points move in the same direction (cosine similiarty)
- Robustness
  - SegColinearity, is looking at a model INSTANCE (each model has all or hld (holding out lmx1b and gdf3)) and then looking at how for the same pert they move in the same direction.
  - Aligned MSE using the quaternion implementaion of Kabsch aligning algorithm we look at how the MSE between like perturbation for a model instance is.
    - This is very stringent but a good measure.
  - Initial MSE
    - How without aligning the MSE
    - used as an internal check to make sure Alignmen tis working AND to see if alignment is actually necessary.
- Classification:
  - F1 Score a measure of how good model is at classification
    - for more info see previous param sweep (referenced above)

Conclusion

Chosen Model (74):

I think Model 74 (or another model in its class should be choosen for these reasons)

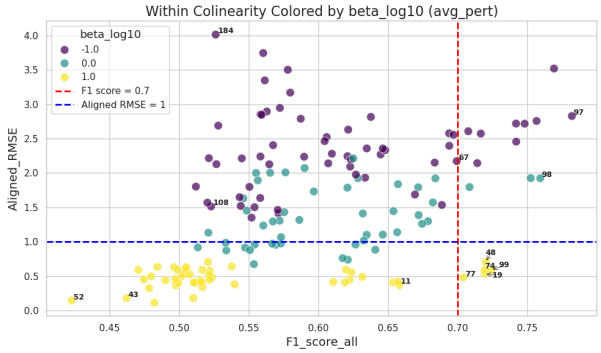
> ## Summary of why model 74
>
> - I choose model 74 because it still has tree like structure, is stable, and it has good classification accuracy
> - Reading below we see that it does a better job extracting data than other models e.g. model 97 (not neccessarily those in its same tanche e.g. 77)
> - It even does better than the previous model we were using!!!
>
> ### Sel Model_Index 74
>
> 
>
> 
>
> 
>
> 
>

Tradeoff of Prettiness vs Classificaion

- There is a tradeoff on prettiness vs classification accuracy, though this is slight.
  It is MOST definitely worth the small decrease in classification accuracy to get a more intepretable morphology space.
  - See MorphSeq Param Sweep with Splines;Dec-17-2024 > Tradeoff of Prettiness vs Classificaion for more detail

Beta Drives Prettiness AND Robustness

- It can be seen that the Beta variable drives prettiness and robustness
- Higher values of the beta variable makes the model make the individual dimensions in the embedding space more iid (independent)
  - It also clearly has the property of making morphology space prettier
  - potentially by smoothing things out.



- This is a surprising result as prettiness/tree likeness is NOT garunteed to be correlated with robustnes/stability.
- High beta model are so rubst that they actually dont need to be aligned! (tgough you should still align them lol)

New Spline Measurement do capture things we wanted them too

- Implementing splines allowed us to finally get at what was the drivers of prettiness
  - previously not able to

Concerns

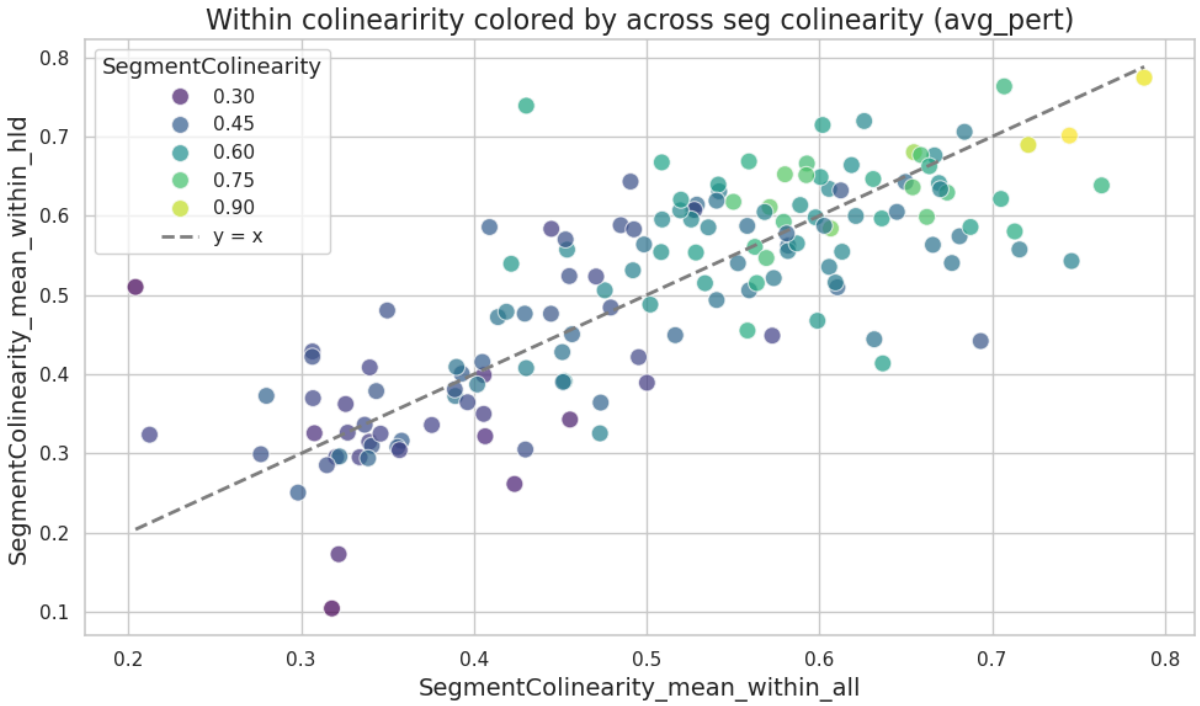
- sometimes splines dont work, but this is when they arent pretty (tight turns, not tree shaped) see model 98
- but these might be inflating the ALigned MSE values but not really worried about this.
  - because when they fail its because data istn tree like.

Next Steps

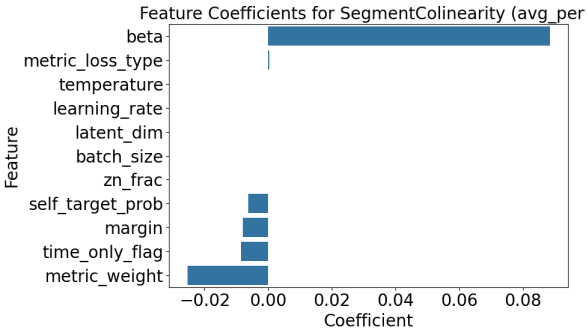
- Consider other measurement that dont need splines (none come to mind)
- I see no obvious change, but if were to make the dispersion_metric first_n and lastn could be calculated from one linear equation, the y intercept oculd be first_n and the multiple the slope by the last index (wich is the dis cofeffiicent here actually) and then add the slope to get last _n
- Of course we need to find way of projecting points onto the spline and then time across it
  - measure variance accross spline
  - measure variablity of time across spline

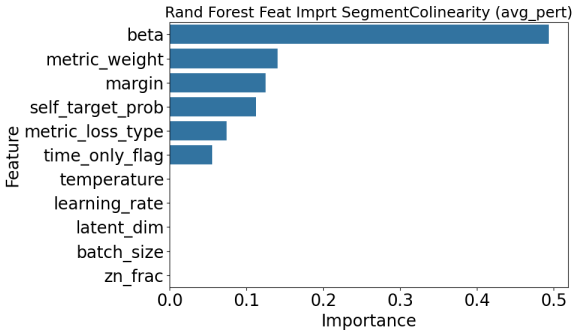
Collinearity

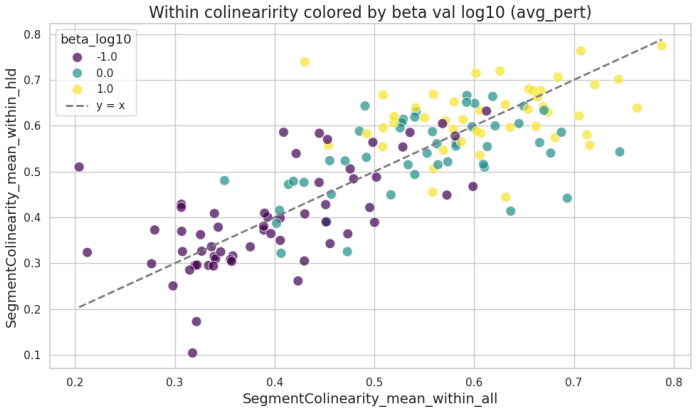
- Here we see that if a model is not colinear within a dataset it wont be in the other,
- Furthermore, that if it has a high colinearity within one then it will be colinear with the held out data! (this can be seen by colorying the points by ACROSS dataset colinearity). This is the first actual evaluation of
- this means that if you project into a space that is alrerady linear, you will get linear manfinolds.

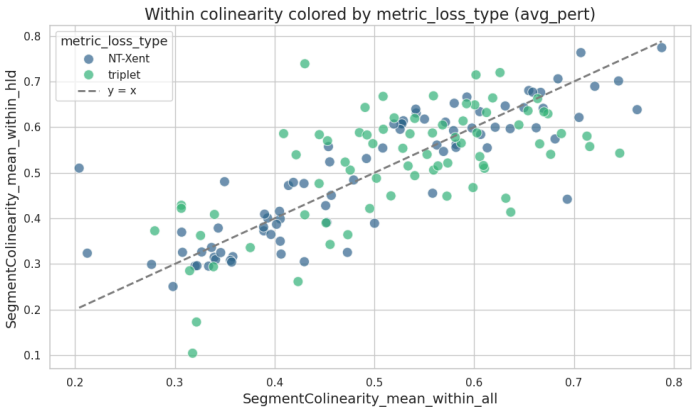


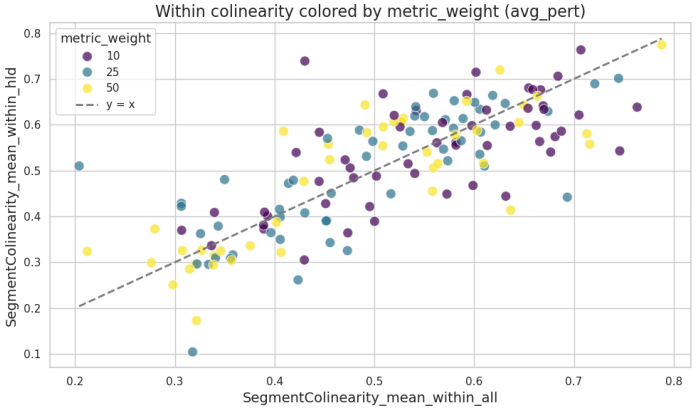
beta as causative for collinearity AND Robustness

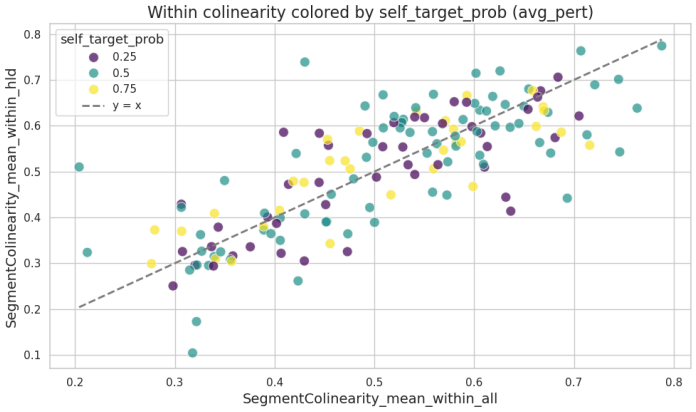






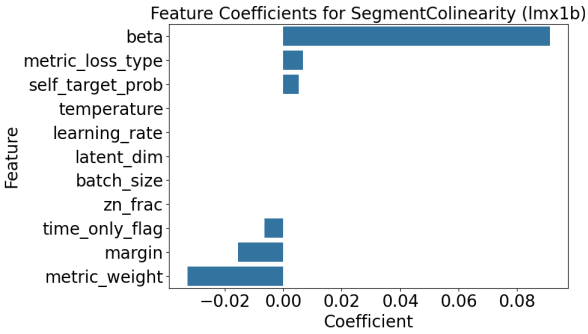


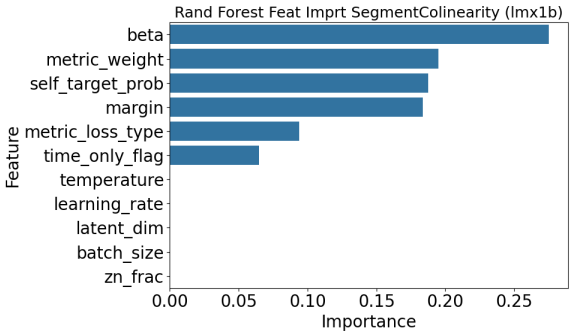




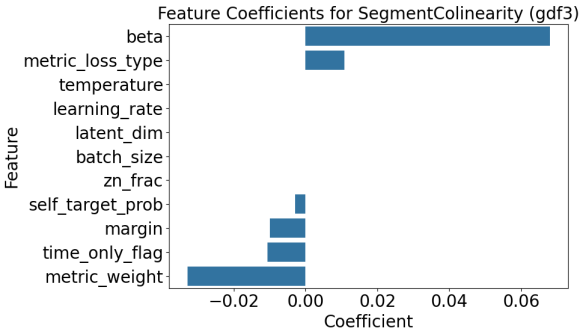
- note that NT-Xent has tighter distribtuions around y=x (more stable)

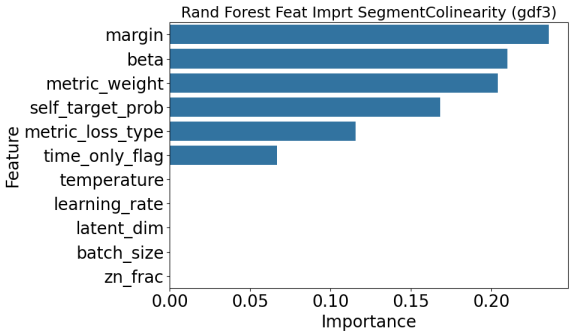
- lmx1b



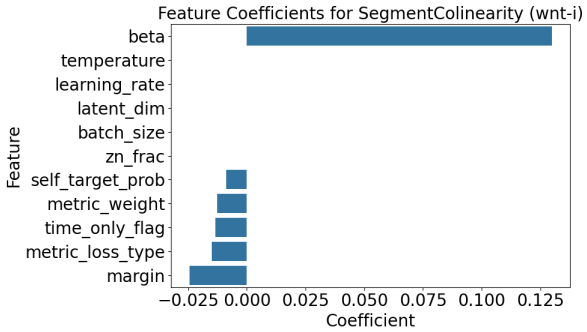


- gdf3



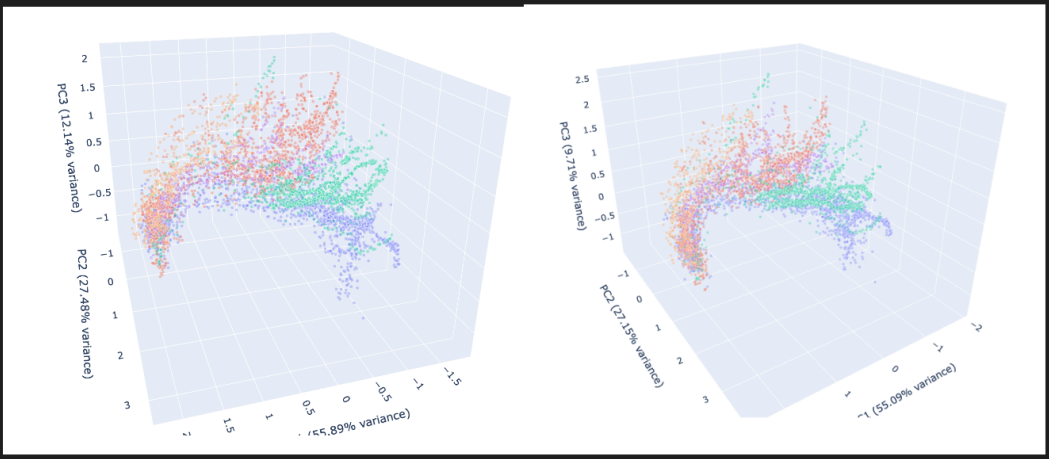


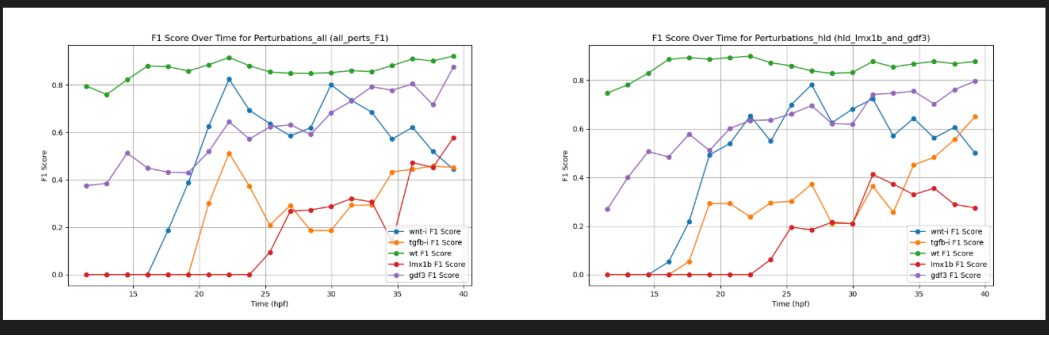
![[Projects/morphSeq/Journal/Dec-2024/attachments/MorphSeq Param Sweep with Splines;Dec-17-2024/image 15.png|300][Projects/morphSeq/Journal/Dec-2024/attachments/MorphSeq Param Sweep with Splines;Dec-17-2024/image 16.png|300]]



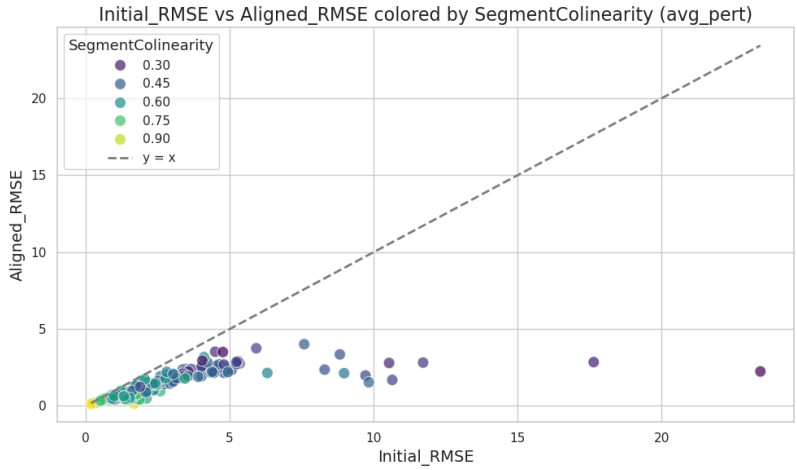
MSE Initial -> Aligned

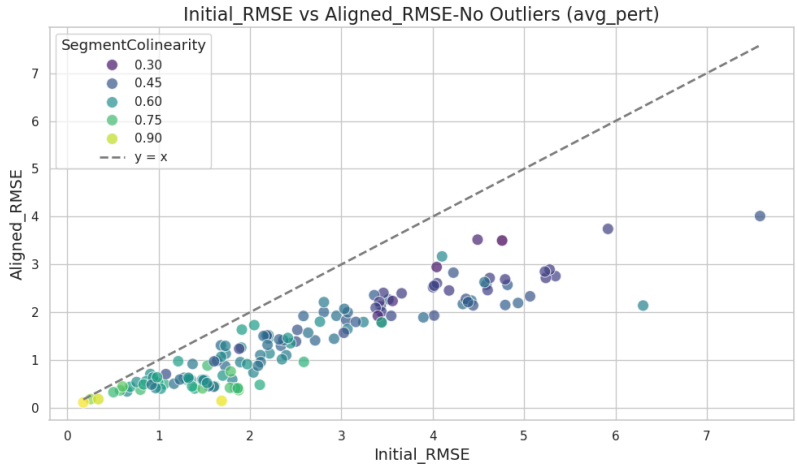
- Here I Noticed that for one of the models that did the best in terms of colinearity metrics, that the PCA was RIGHT on top of eachother pre alignemnt which was not garunteed at all. (this would imply that PCA is generating the same eignevectors acorss training regimes)
- example 71 is a prime example!
    - MorphSeq Paramater Sweep Takeaways;Nov-28-2024

- **71: F1 0.72**
  - time onlywas used
  - 

  - 

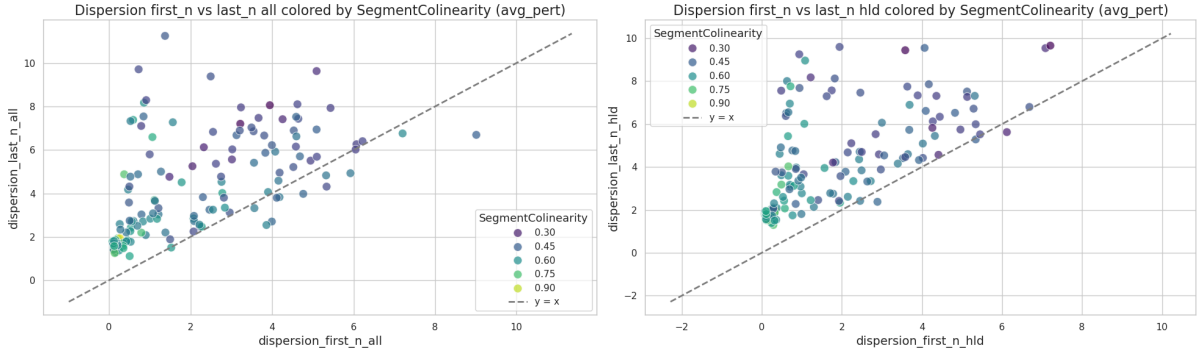
- This is confirmed here! that yes if the model is colinear then it will already by aligned!
  - Note that all the lines are under the y=x line which means Initial RMSE < Aligned RMSE
  - (second plot is removeing outliers)

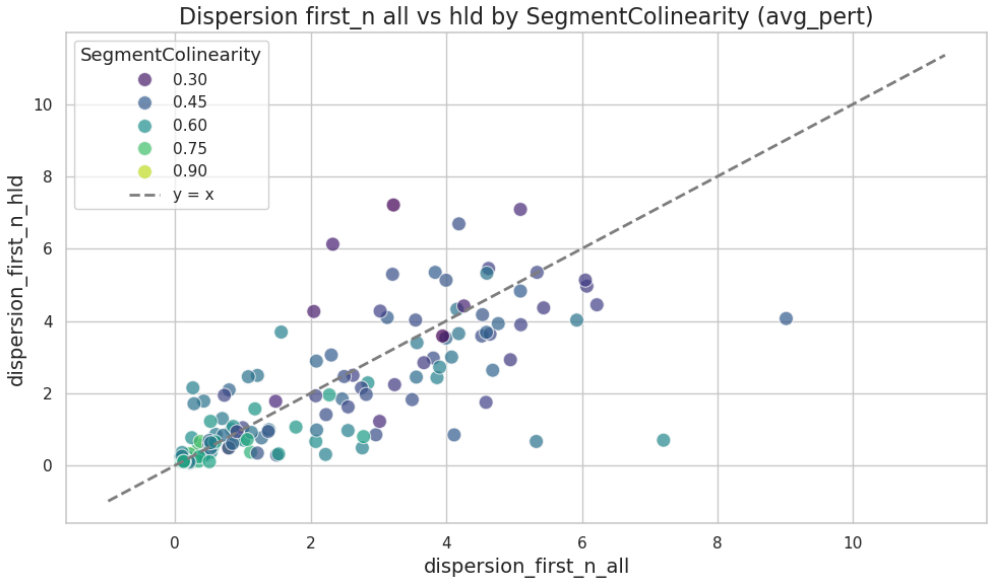




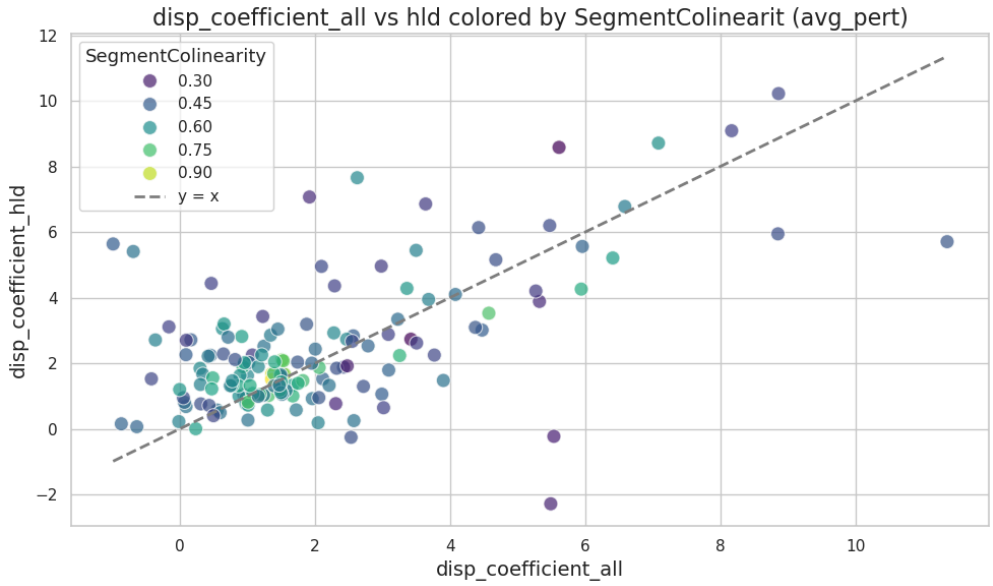
Dispersion (tree likeness)

- we wanted to see how tree like our models by looking at the first n and last n points and seeing hoe spread apart they are, we want firs n to be close to zero and last n to be bigger. Additionally we use dispersion coefficient which simply looks at dispersion along the spline. higher dispersion_coef measn higher speard across the spline. (note n = 5)
- main takeway from these plot is that as long as the model is average or better at the segment collinearity metrics then it will will be tree and that generally dispersion increases over the spline.
- here we see that for models that do ok with linearity metrics have a first_n close to 0 AND last n around 2
    - also note that when held out model the beginning points (first_n) become more spread out as well





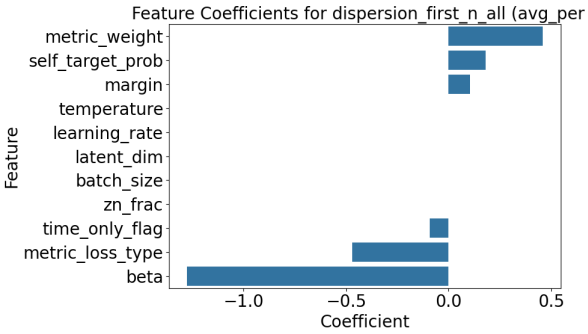
- We can see that model becomes slightly more disperse (higher dispersion coefficient in hld) when holding out perturbations.

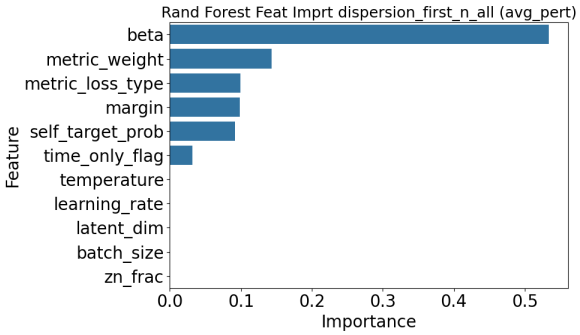


---

drivers of Dispersion:

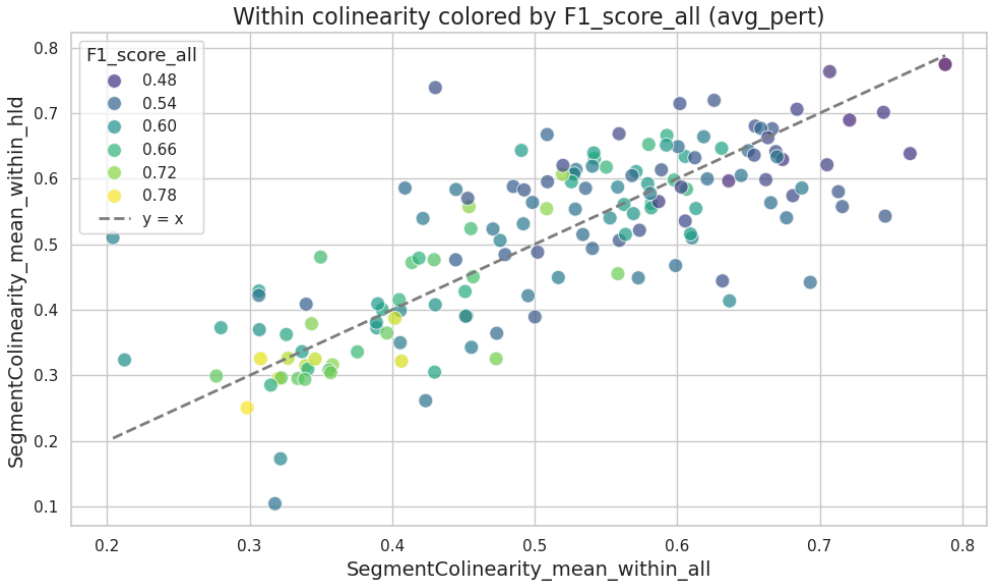
- Higher beta --> more linear --> closer first_n --> more tree like



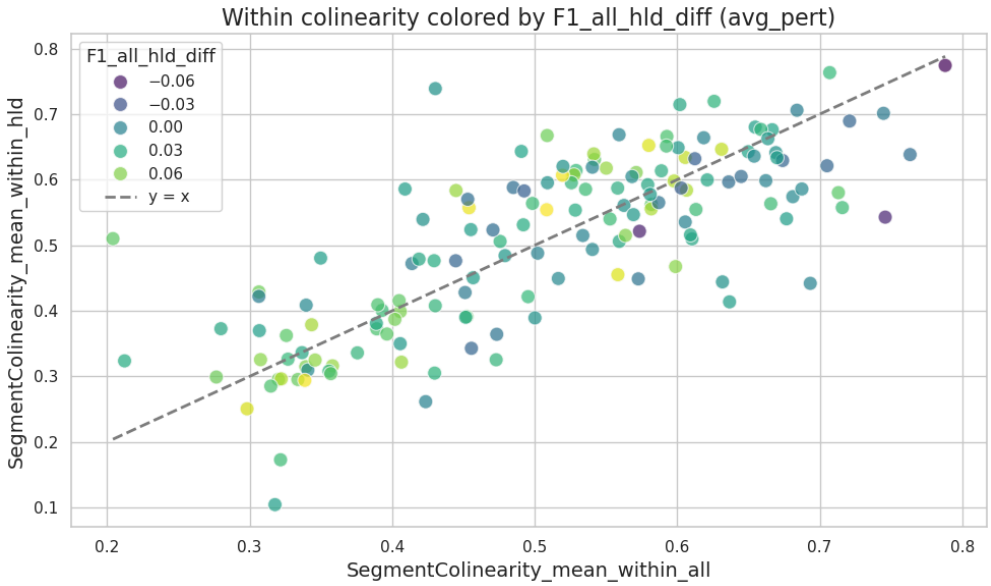


Prettiness vs Classification Accuracy

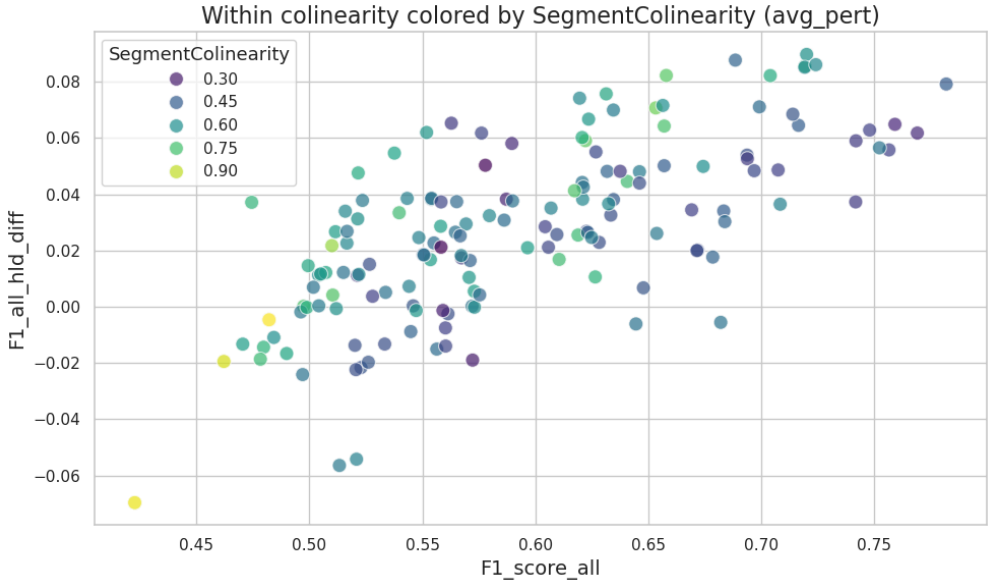
- clear trend that:
  1. classification can come at the cost of prettiness and vice versa
  2. for a given f1 score more colinearirity decreases classication robustness
- coloring segcoliniarity by F1 score showes that there is generally a tradeoff that lower F1 scores (bluer) corresponds to higher colinearity metrics



- Hard to draw a trend between F1 score difference (robustness) and seg colinearity

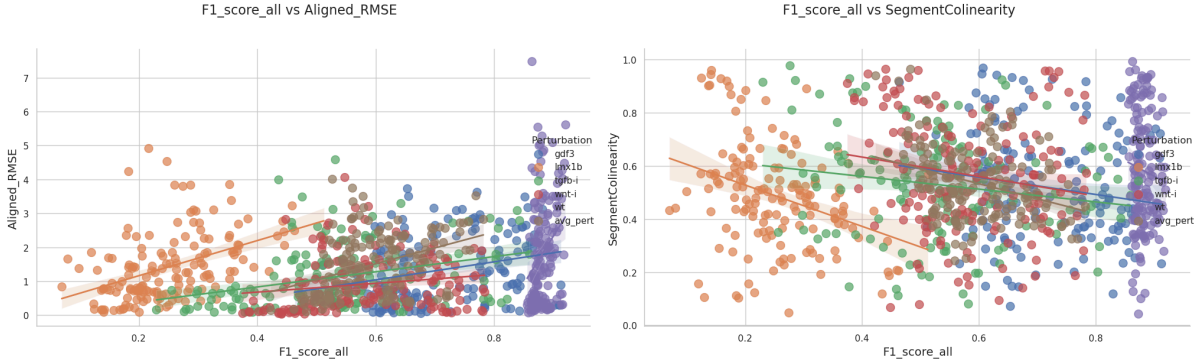


- Slight signal but you can SEE that for a given F1 score higher colinearity means that there wil be more of a perforance loss wrt. F1 score robustness...
  - though the tradeodd isnt anything crazy...



Classification on per Pert level and Prettiness

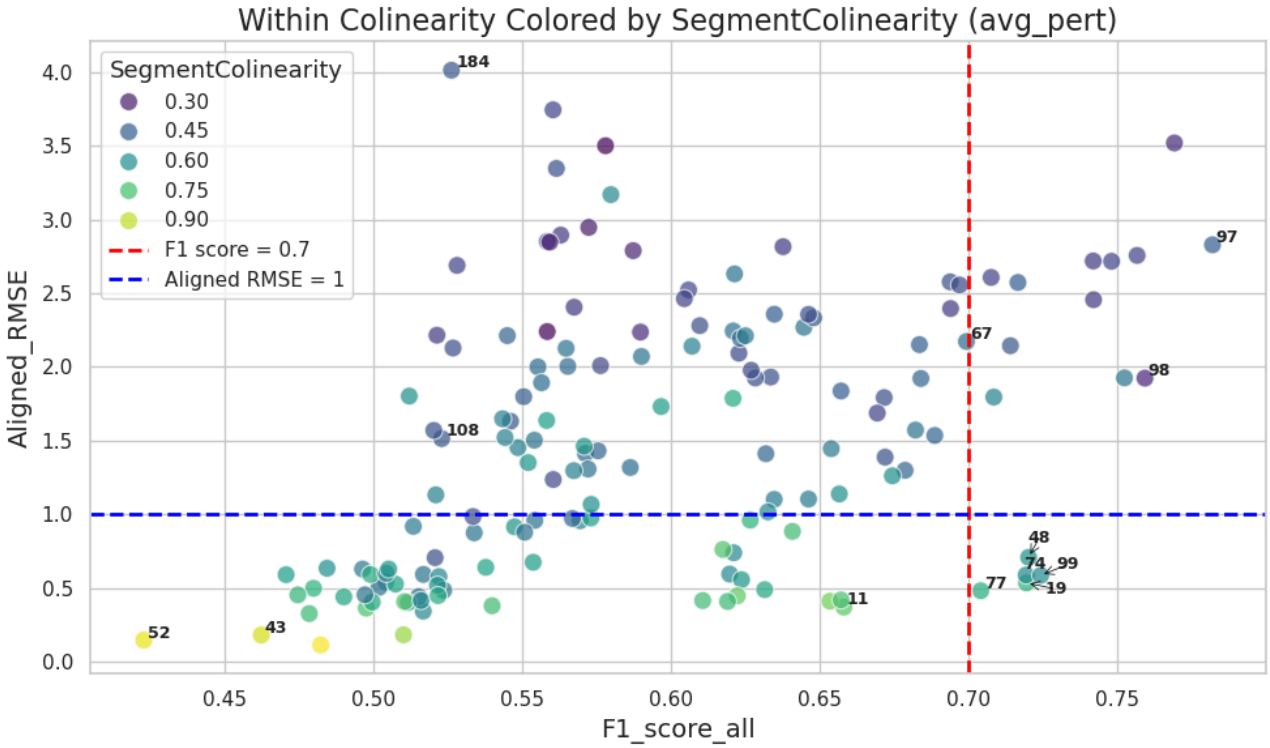
- Trends are recapitulated, tradeoff between prettiness and classification



Picking models

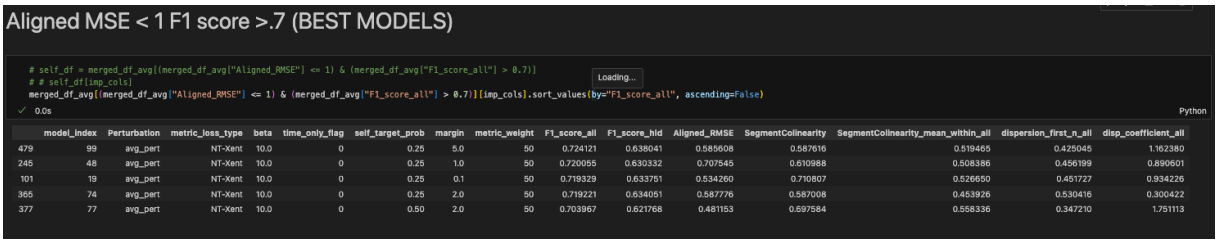
First pass with metrics we care about F1 and Align MSE (prettiness)

- I think having **F1 score > .7 and Aligned MSE < 1** are good acceptable models (look at blue and red lines)
- We care about F1 score (classification), Aligned RMSE (robustness AND pretiness), I am coloring by other things too, to make sure they have good properties.
- Note we know Aligned RMSE is associated with robustness AND pretiness) because of [MorphSeq Param Sweep with Splines;Dec-17-2024 > MSE Initial -> Aligned]



Sel Models from different parts of the plot used to select models:

```
using Aligned RMSE and F1 score as discriminators
```

- Best models (98,48,19,74,77)
- 

- Second Tranche (97,98)
  - Aligned MSE >1 F1 score >.7 (Subset)

```
merged_df_avg[(merged_df_avg["Aligned_RMSE"] > 1) & (merged_df_avg["F1_score_all"] > 0.7)& (merged_df_avg["model_index"].isin([97,98]))][imp_cols].sort_values(by="F1_score_all", ascending=False)
```

[7] ✓ 0.0s Python

	model_index	Perturbation	metric_loss_type	beta	time_only_flag	self_target_prob	margin	metric_weight	F1_score_all	F1_score_hld	Aligned_RMSE	SegmentColinearity	SegmentColinearity_mean_within_all	dispersion_first_n_all	disp_coefficient_all
467	97	avg_pert	NT-Xent	0.1	0	0.25	5.0	50	0.781858	0.702657	2.827774	0.447553	0.298167	9.013367	-0.986660
473	98	avg_pert	NT-Xent	1.0	0	0.25	5.0	50	0.759139	0.694277	1.923200	0.325243	0.406500	1.483292	2.308945

- Third Tranche (11,43,52)
  - Aligned MSE <=1 F1 score <.7 (Subset)

```
merged_df_avg[(merged_df_avg["Aligned_RMSE"] <= 1) & (merged_df_avg["F1_score_all"] < 0.7) & (merged_df_avg["model_index"].isin([52,43,11]))][imp_cols].sort_values(by="F1_score_all", ascending=False)
```

[7] ✓ 0.0s Python

	model_index	Perturbation	metric_loss_type	beta	time_only_flag	self_target_prob	margin	metric_weight	F1_score_all	F1_score_hld	Aligned_RMSE	SegmentColinearity	SegmentColinearity_mean_within_all	dispersion_first_n_all	disp_coefficient_all
59	11	avg_pert	NT-Xent	10.0	0	0.25	0.1	25	0.657936	0.575697	0.369914	0.779365	0.580015	0.146752	1.528816
215	43	avg_pert	NT-Xent	10.0	1	0.50	1.0	25	0.462121	0.481579	0.180996	0.920061	0.720897	0.128076	1.363076
257	52	avg_pert	NT-Xent	10.0	1	0.50	1.0	50	0.422659	0.492315	0.146000	0.940644	0.788008	0.256395	1.485208

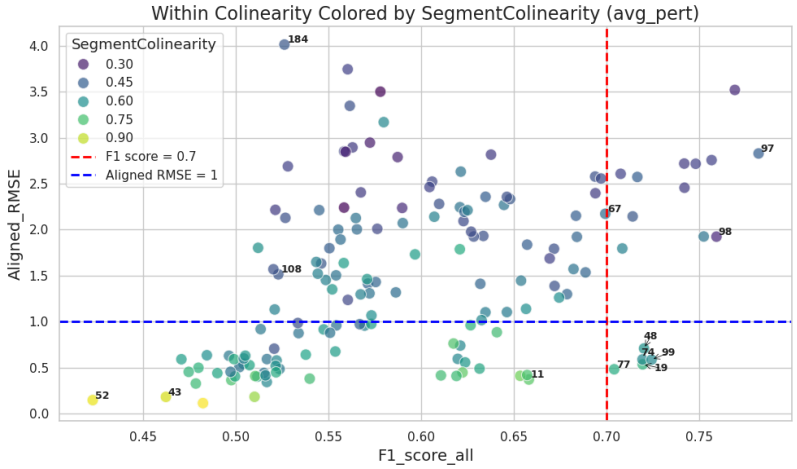
- Fourth Tranche (67, 184, 108)
  - Aligned MSE >1 F1 score <.7 (Subset)

```
merged_df_avg[(merged_df_avg["Aligned_RMSE"] > 1) & (merged_df_avg["F1_score_all"] < 0.7) & (merged_df_avg["model_index"].isin([108,67,184]))][imp_cols].sort_values(by="F1_score_all", ascending=False)
```

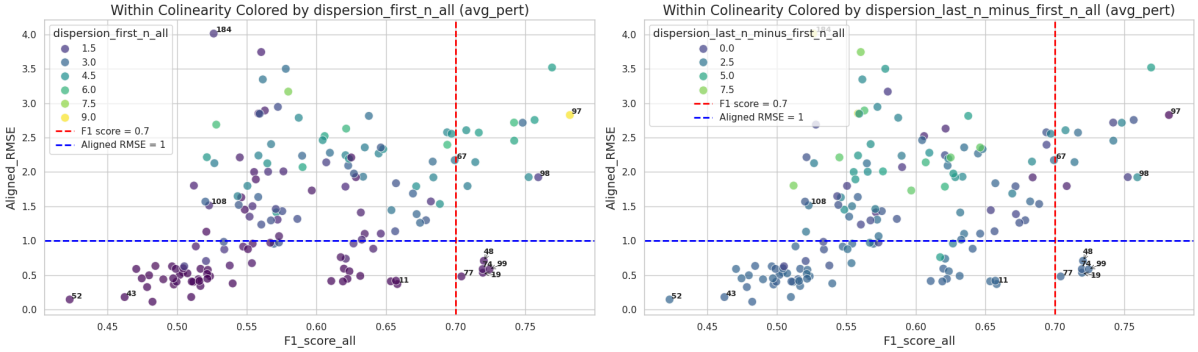
[35] ✓ 0.0s Python

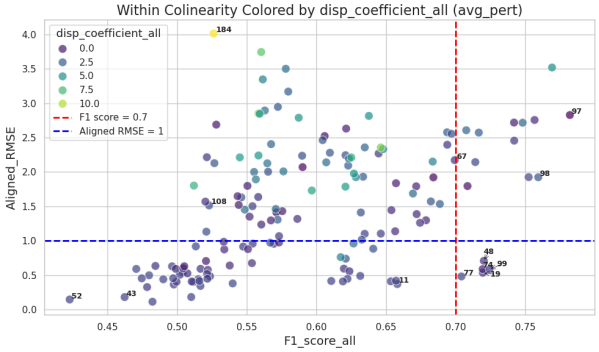
	model_index	Perturbation	metric_loss_type	beta	time_only_flag	self_target_prob	margin	metric_weight	F1_score_all	F1_score_hld	Aligned_RMSE	SegmentColinearity	SegmentColinearity_mean_within_all	dispersion_first_n_all	disp_coefficient_all
323	67	avg_pert	NT-Xent	0.1	0	0.50	2.0	25	0.699149	0.628076	2.171811	0.499992	0.322169	3.557945	1.022507
881	184	avg_pert	triplet	0.1	0	0.25	5.0	10	0.526068	0.545798	4.012054	0.443271	0.444577	1.374297	11.356005
515	108	avg_pert	triplet	0.1	0	0.50	0.1	10	0.522775	0.544393	1.513283	0.435893	0.612368	1.213509	2.058222

- assuring models have nice segment colinearity

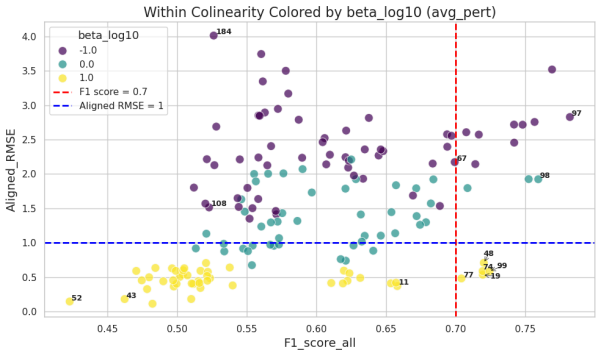


- assuring models have tree like structure low disp_first_n and non abnormal dispersion_coefficient

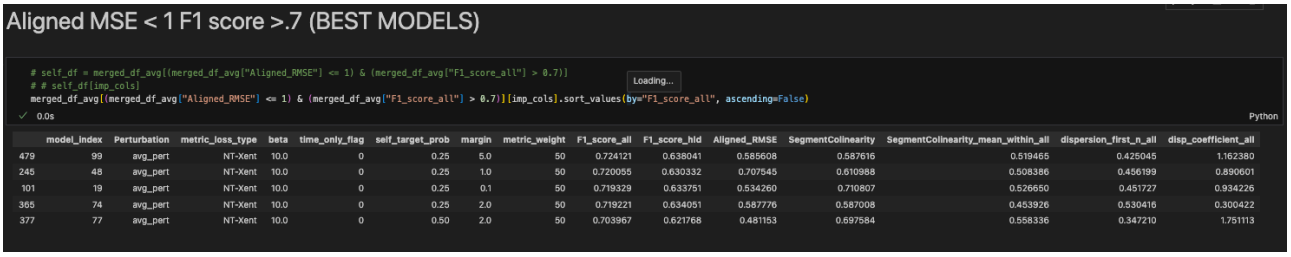




- we can see beta is driving the prettiness

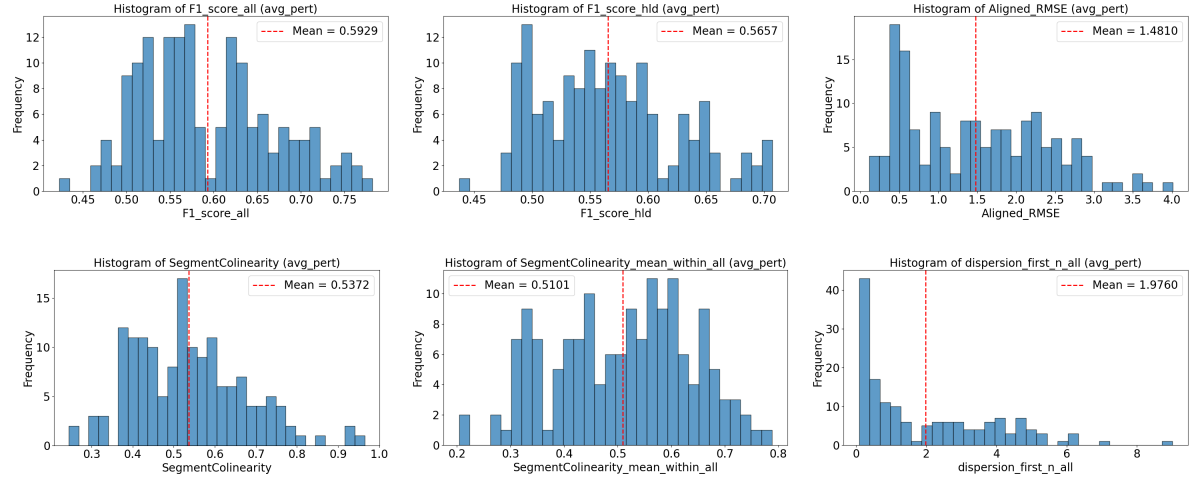


Extracting the top model models (98,48,19,74,77):



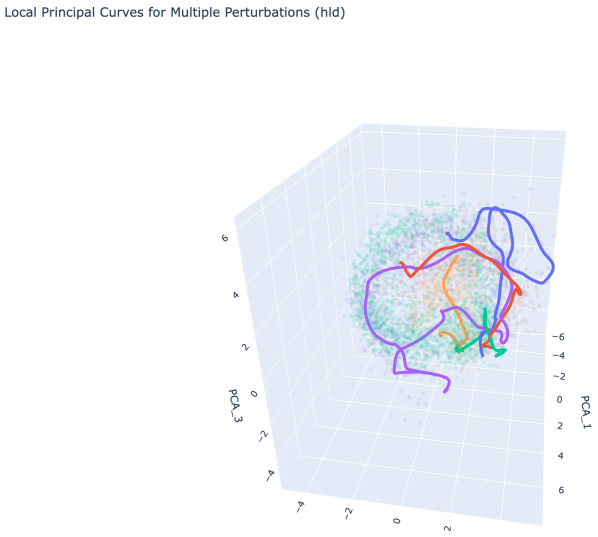
model_index	Perturbation	metric_loss_type	beta	time_only_flag	self_target_prob	margin	metric_weight	F1_score_all	F1_score_hld	Aligned_RMSE	SegmentColinearity	SegmentColinearity_mean_within_all	dispersion_first_n_all	disp_coefficient_all
479	99	avg_pert	NT-Xent	10.0	0	0.25	5.0	50	0.724121	0.638041	0.585608	0.587616	0.519465	0.425045	1.162380
245	48	avg_pert	NT-Xent	10.0	0	0.25	1.0	50	0.720055	0.630332	0.707545	0.610988	0.508386	0.456199	0.890601
101	19	avg_pert	NT-Xent	10.0	0	0.25	0.1	50	0.719329	0.633751	0.534260	0.710807	0.526650	0.451727	0.934226
365	74	avg_pert	NT-Xent	10.0	0	0.25	2.0	50	0.719221	0.634051	0.587776	0.587008	0.453926	0.530416	0.300422
377	77	avg_pert	NT-Xent	10.0	0	0.50	2.0	50	0.703967	0.621768	0.481153	0.697584	0.558336	0.347210	1.751113

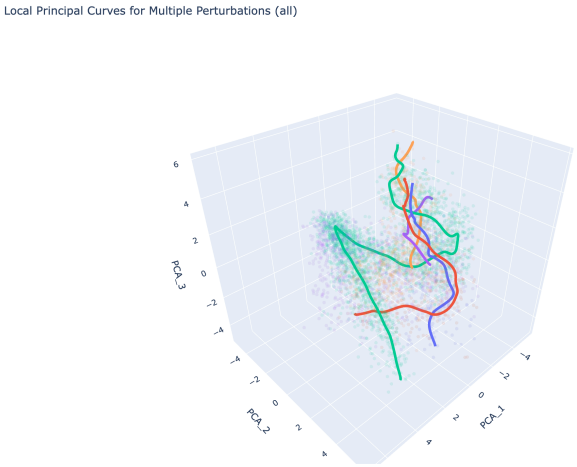
  - None of them use the time only flag, and ALL of them are NT-Xent
- we can see that model is in top quartile for Classification metrics and has very close first_n, and is average in terms of segment colinearity

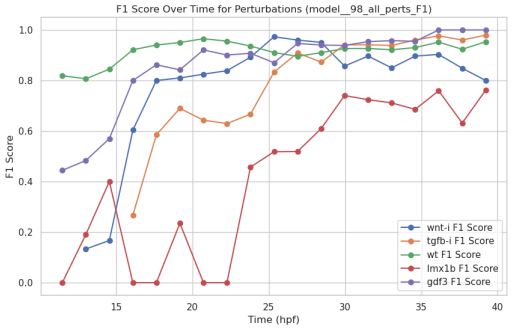


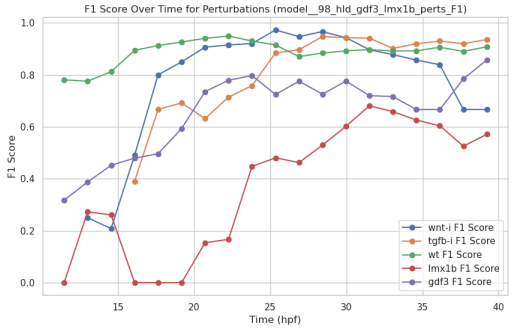
Looking at Best Models (Lower right of plot)

Sel Model_Index 98:

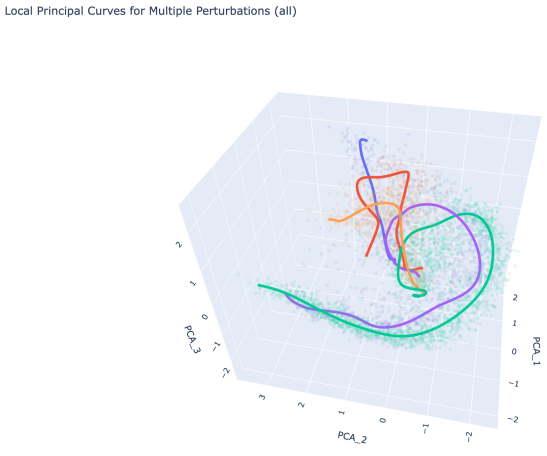


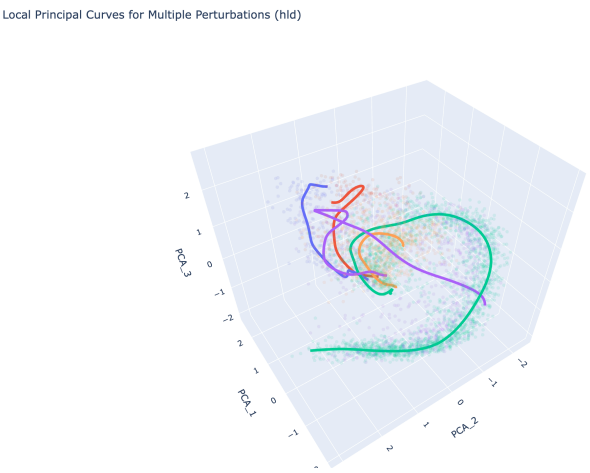


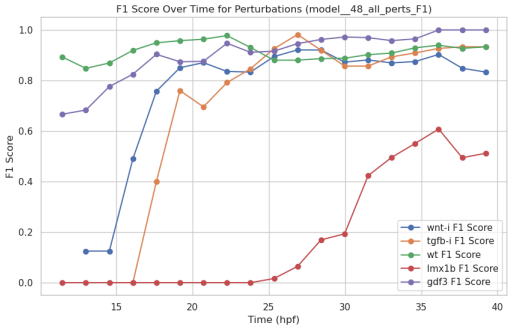


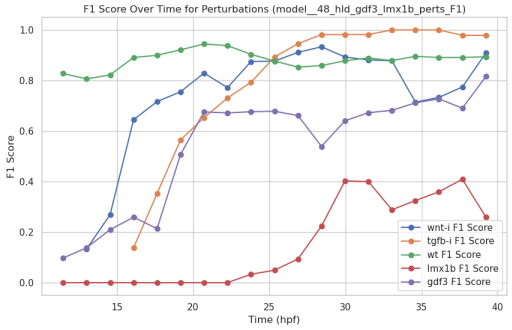


Sel Model_Index 48

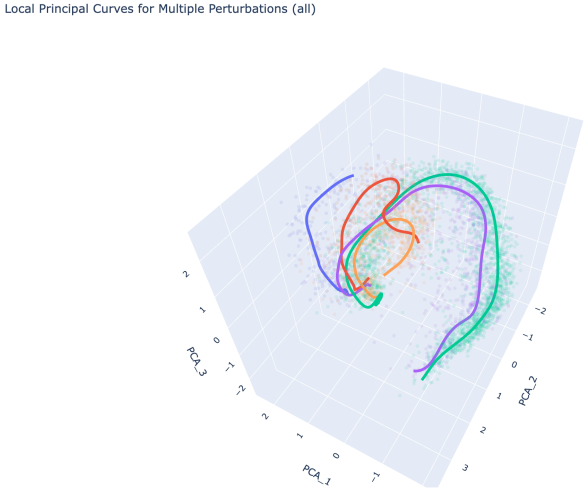


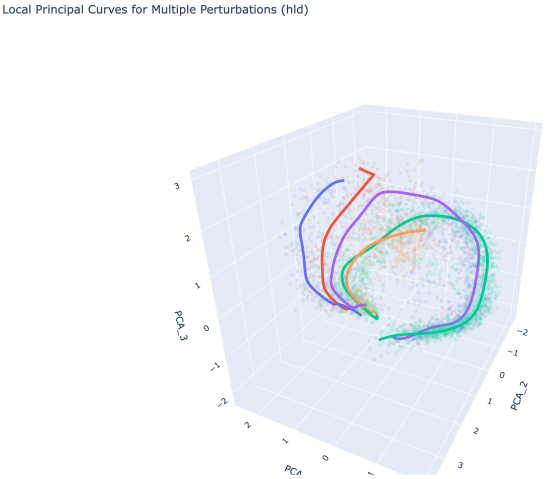


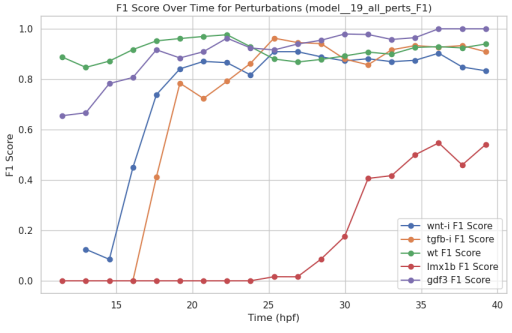


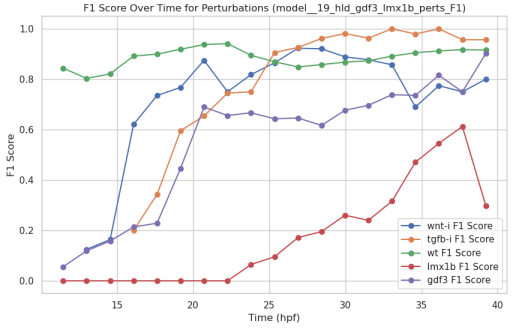


Sel Model_Index 19



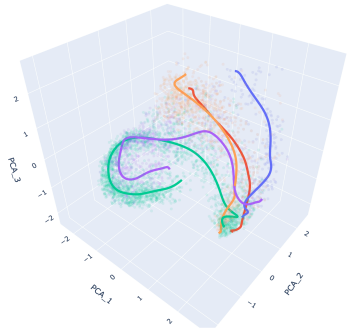




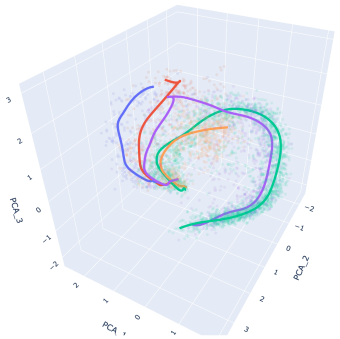


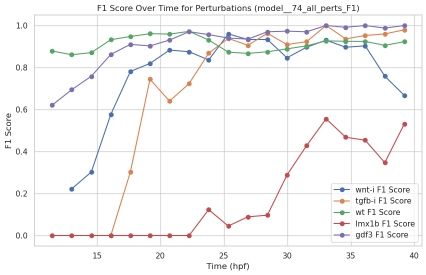
Sel Model_Index 74

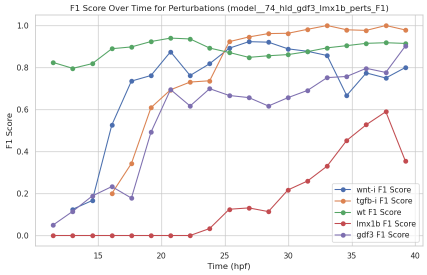
Local Principal Curves for Multiple Perturbations (all)



Local Principal Curves for Multiple Perturbations (hld)

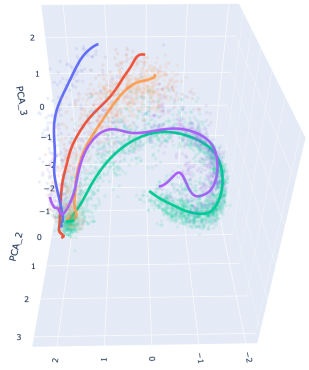




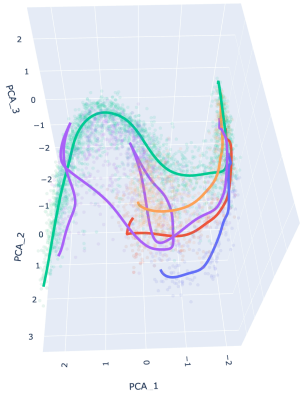


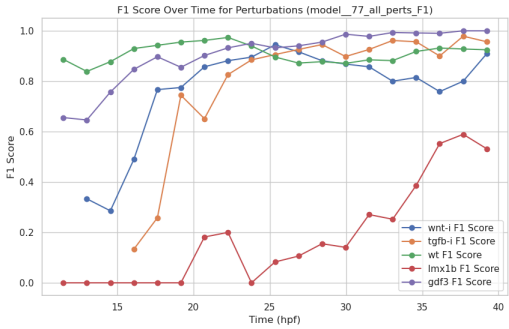
Sel Model_Index 77

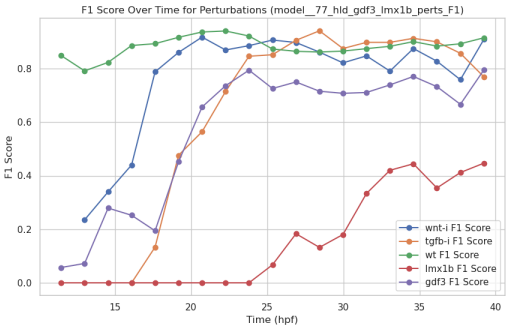
Local Principal Curves for Multiple Perturbations (all)



Local Principal Curves for Multiple Perturbations (hld)





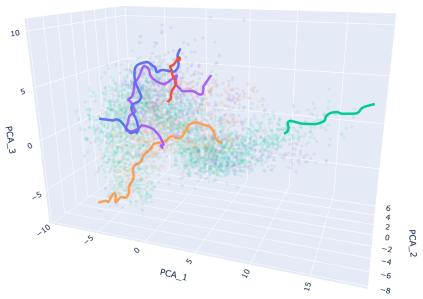


Looking at Second Traunche (97,98; Top Right)

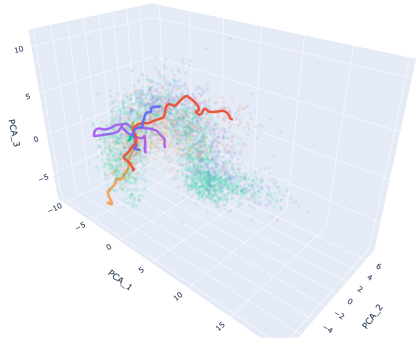
- These models have the highest F1 score 97 has .78 which is the highers

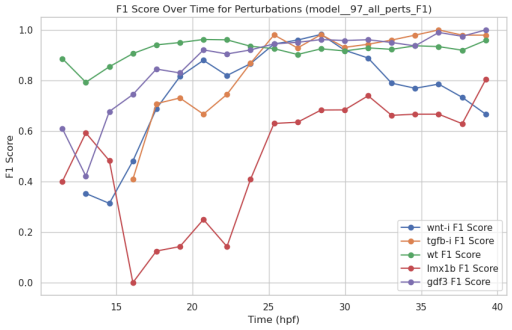
Sel Model_Index 97

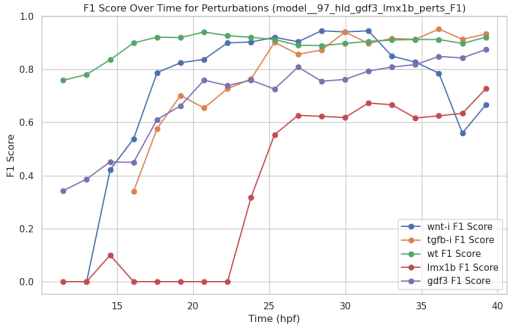
Local Principal Curves for Multiple Perturbations (all)



Local Principal Curves for Multiple Perturbations (hld)

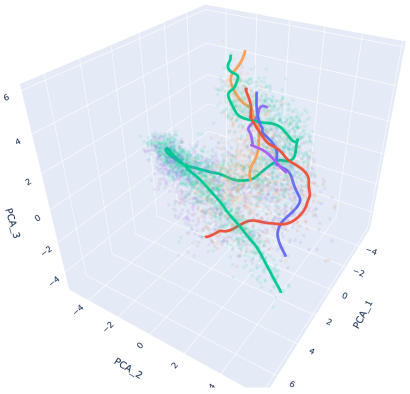




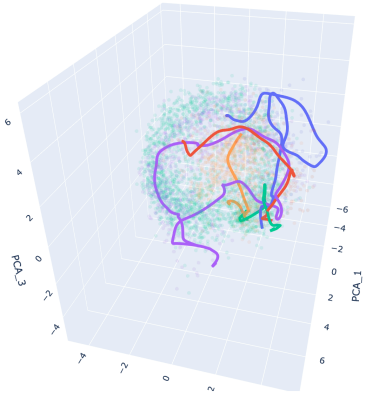


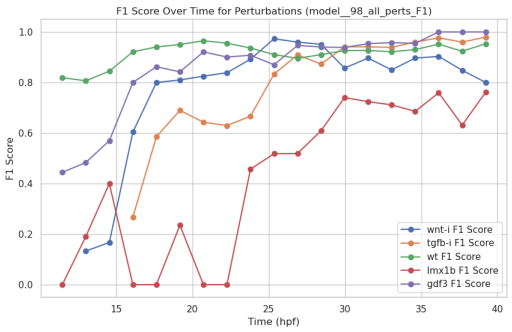
Sel Model_Index 98

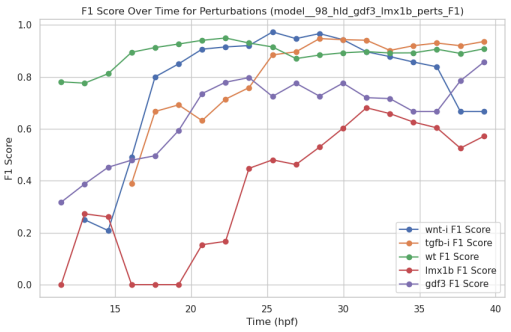
Local Principal Curves for Multiple Perturbations (all)



Local Principal Curves for Multiple Perturbations (hld)

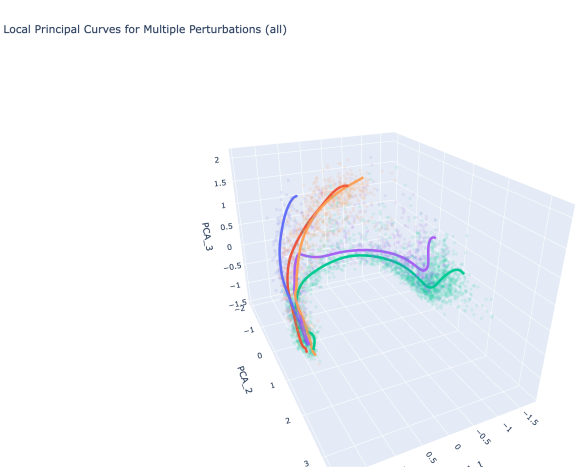


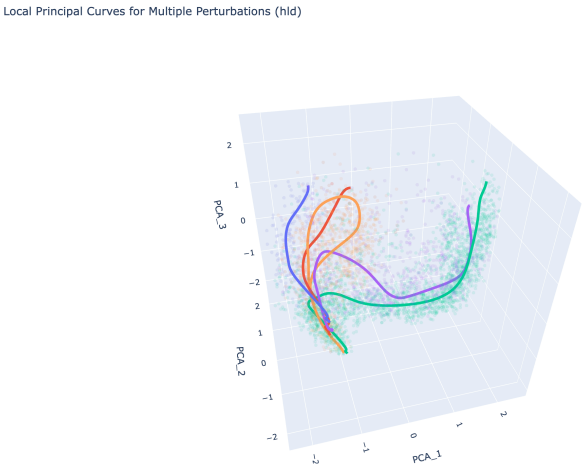


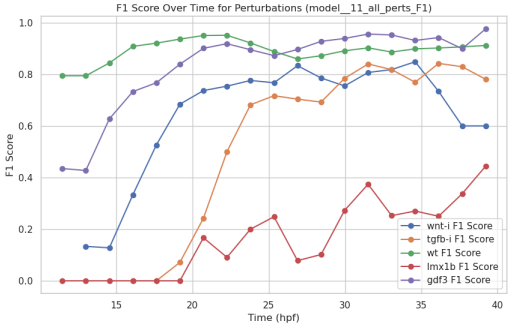


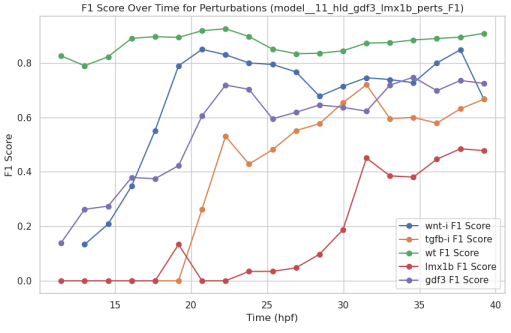
Looking at Third Tranche (11,43,52; Lower left)

Sel Model_Index11

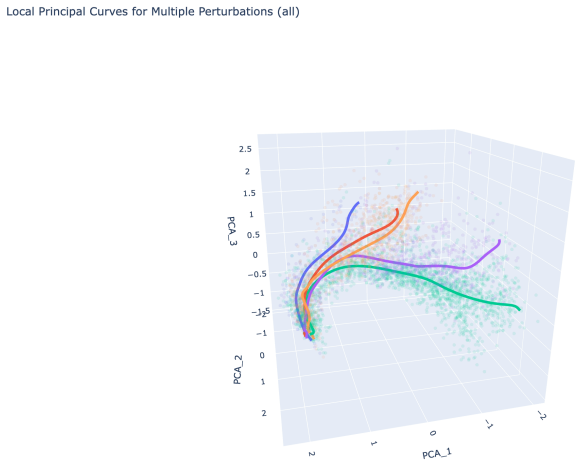


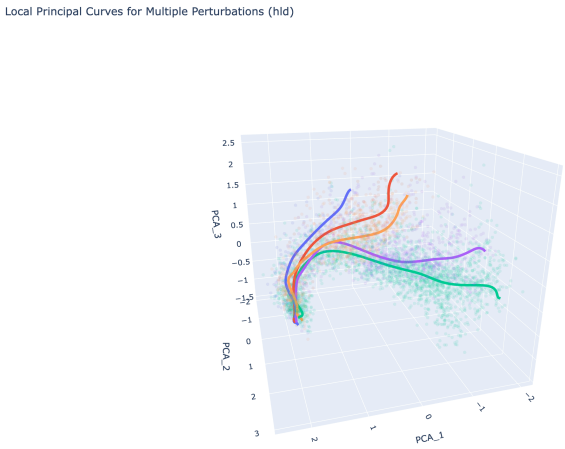


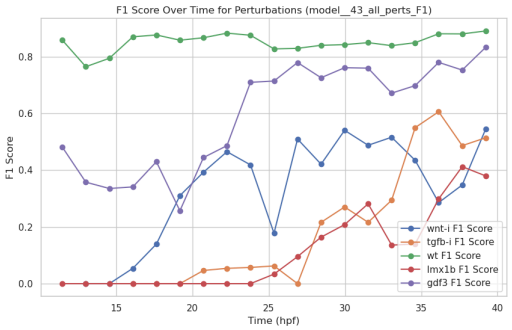


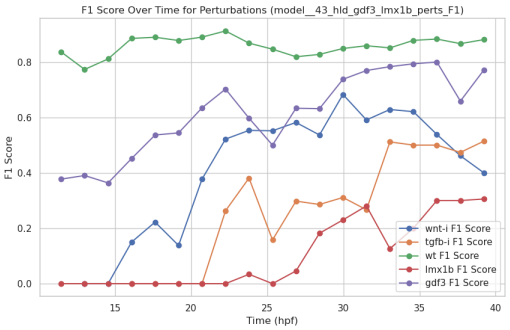


Sel Model_Index 43

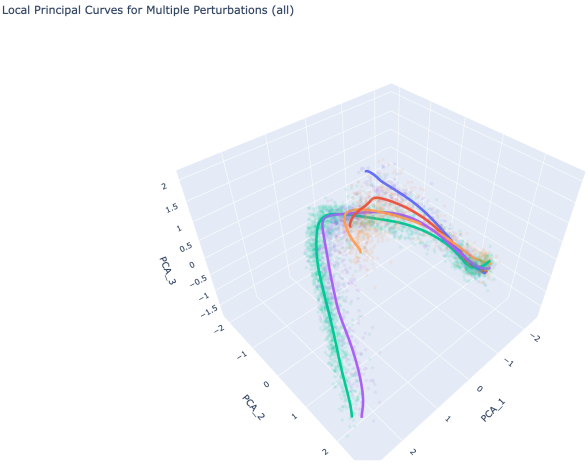


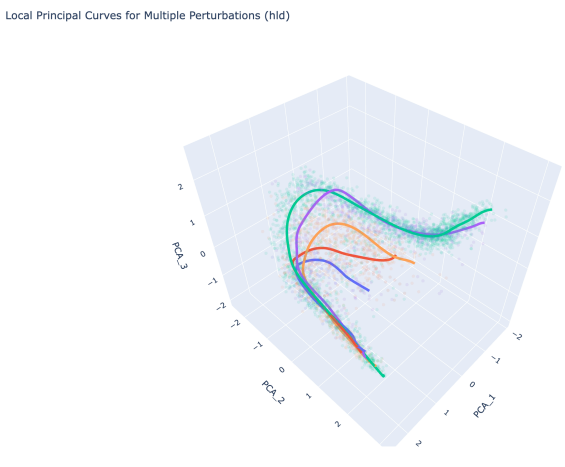


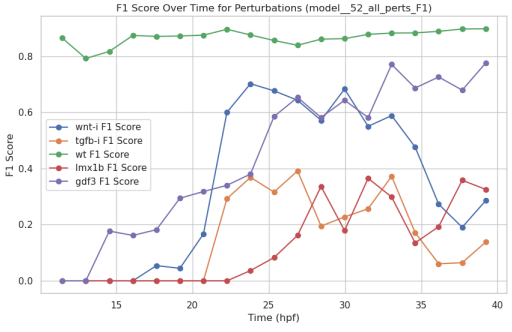


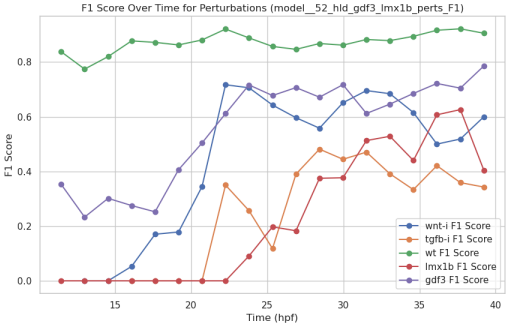


Sel Model_Index 52



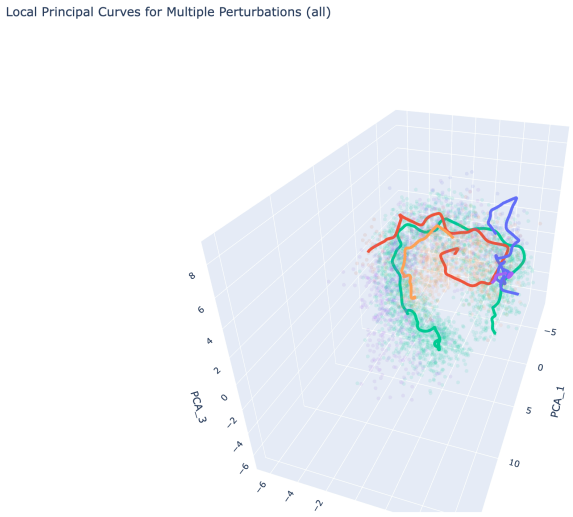


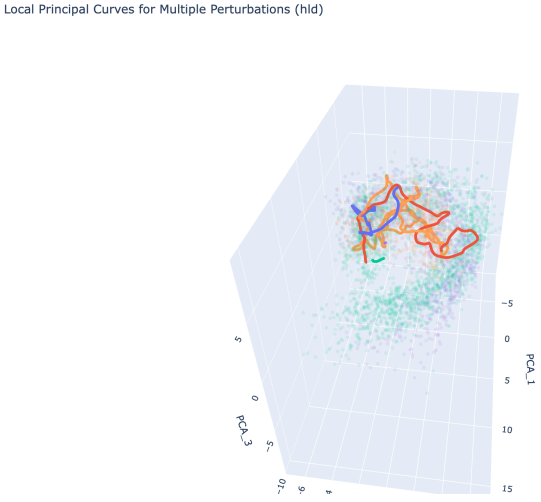


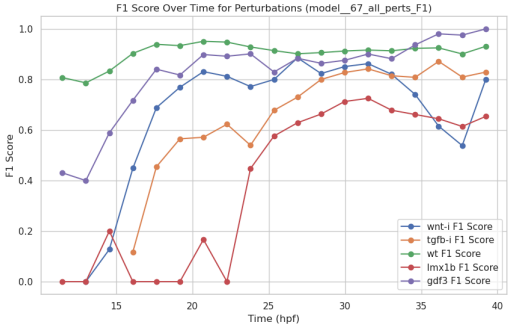


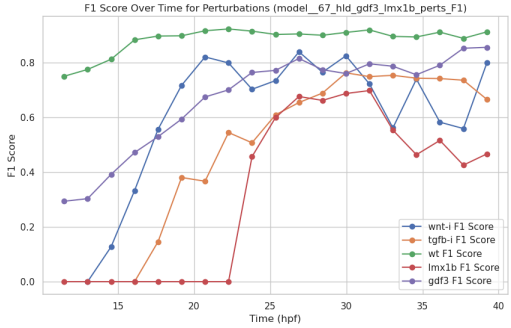
Looking at Fourth Tranche (67, 184, 108; Top Left)

Sel Model_Index 67

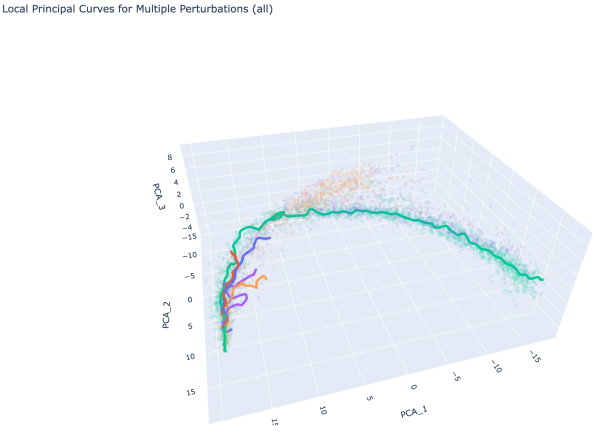


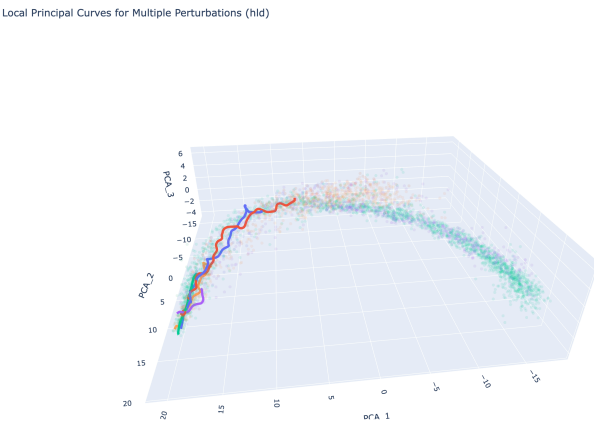


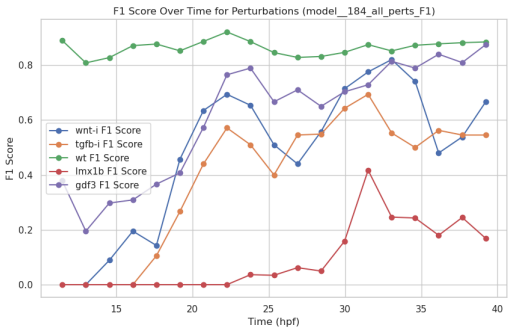


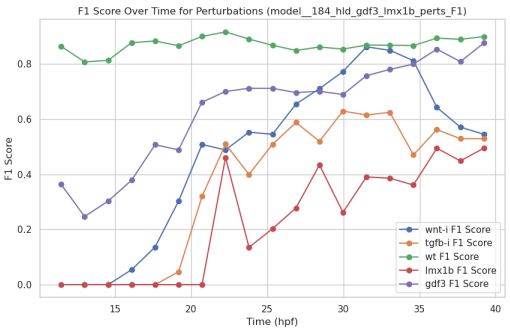


Sel Model_Index 184

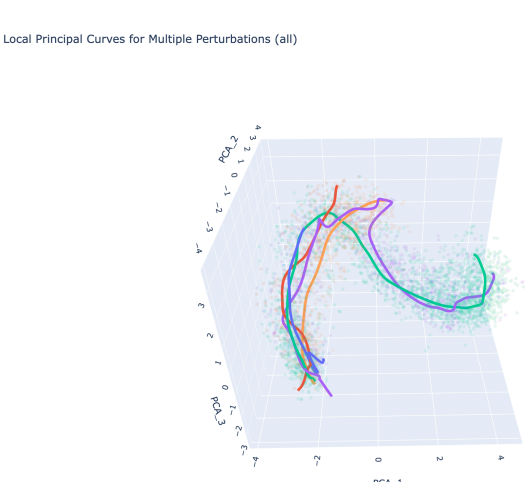


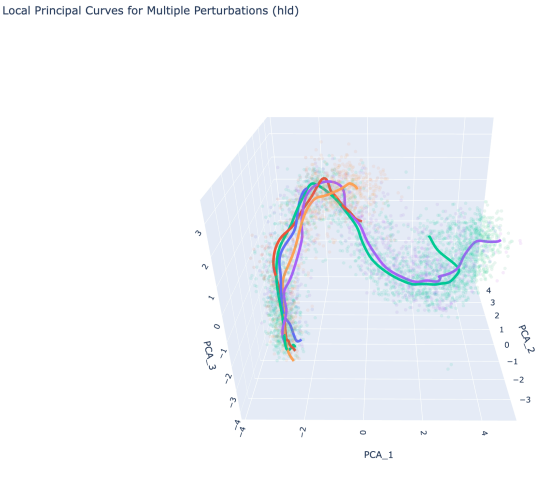


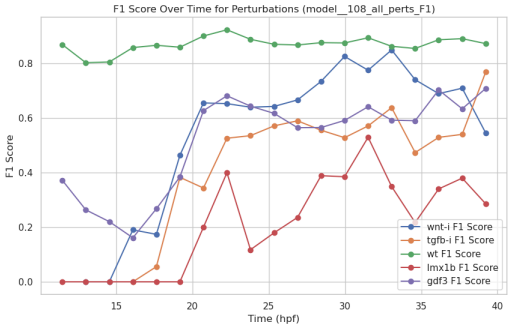


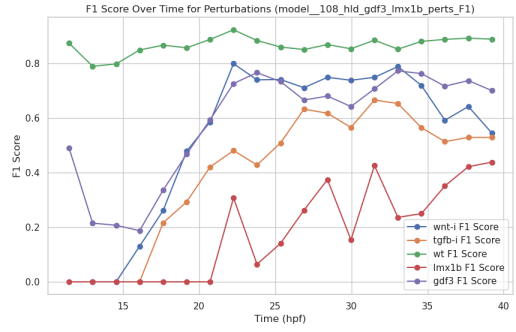


Sel Model_Index 108





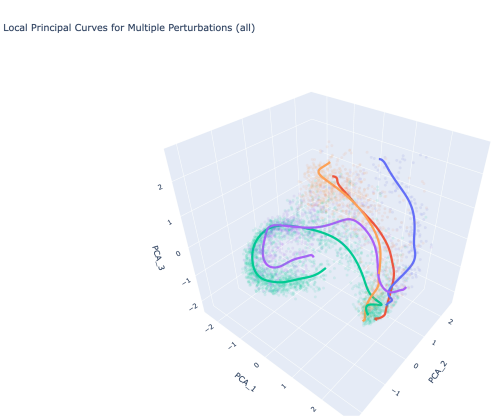
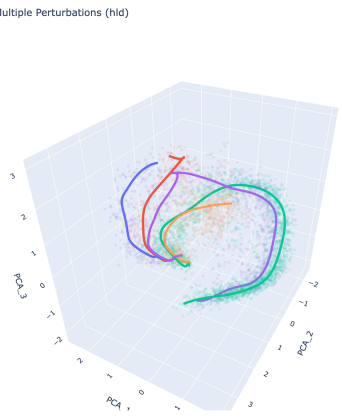
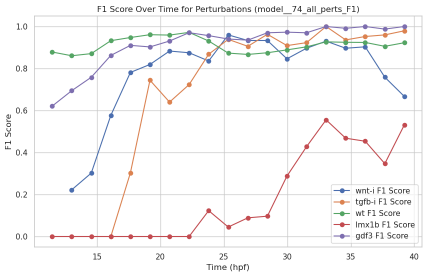
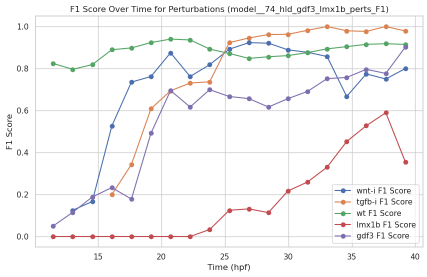




Choosing Model 74

Summary of why model 74

- I choose model 74 because it still has tree like structure, is stable, and it has good classification accuracy
- Reading below we see that it does a better job extracting data than other models e.g. model 97 (not neccessarily those in its same tanche e.g. 77)
- It even does better than the previous model we were using!!!

> ## Sel Model_Index 74
>
> 
>
>
> 
>
>
> 
>
>
> 
>

comparing it to models with higher classification [MorphSeq Param Sweep with Splines;Dec-17-2024 > Looking at Second Traunche (97,98; Top Right)]

Pretinness:

- Obviously these models with classifcation just have more compact less and less tree like morphology space (notice how the perts dont branch out)
  - Model 74 wins this easily
  Now Classification is more nuanced but it still LOSES:
- The model 97 (and 98) seem to really be edging out the improved F1 scores through improvements in lmx1b.
- However theres two things that should be noted:

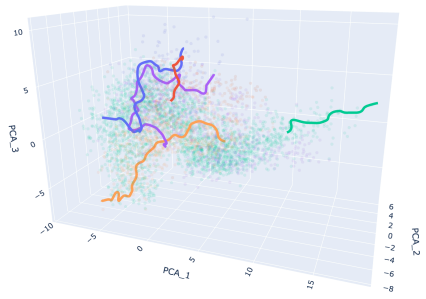
1. That the gains in lmx1b in earlier time points arent learned when held out
2. In all Perts other than Lmx1b the model 97 is doing WORSE

- These two observations make me suspicious that it is learning non-biological information. If the model couldve repeated this performance in the hld I wouldve beleived it was possibly learning something useful.

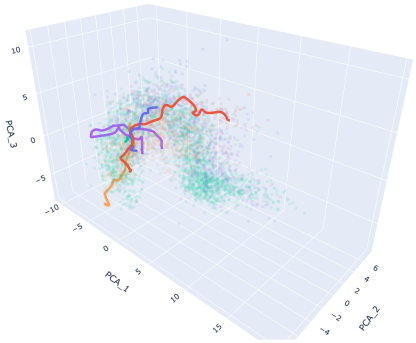
Model 97

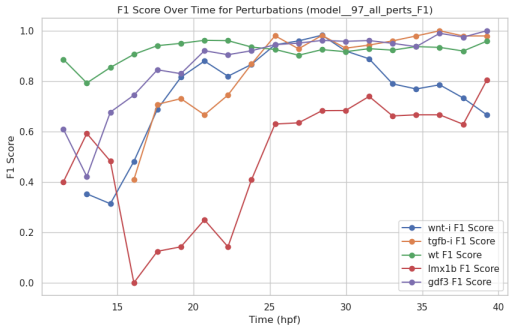
Sel Model_Index 97

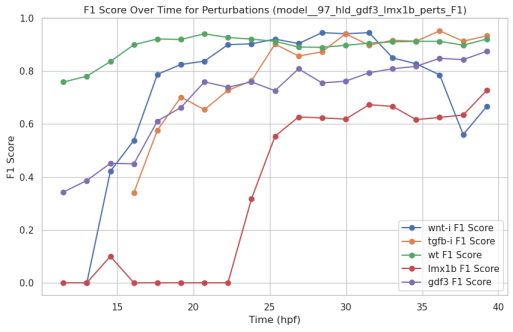
Local Principal Curves for Multiple Perturbations (all)



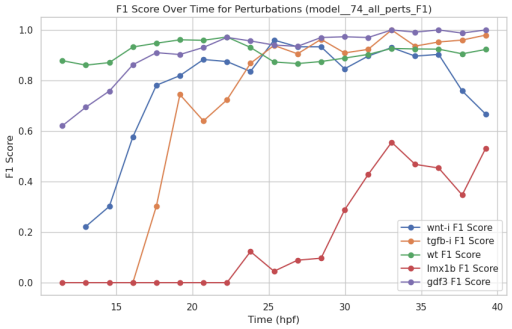
Local Principal Curves for Multiple Perturbations (hld)

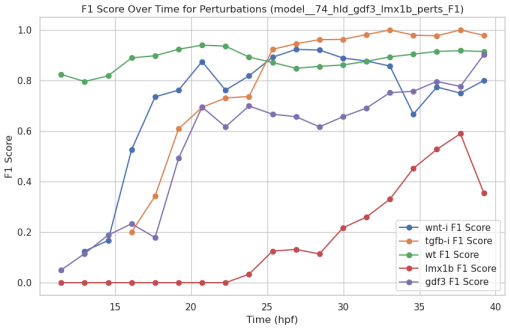






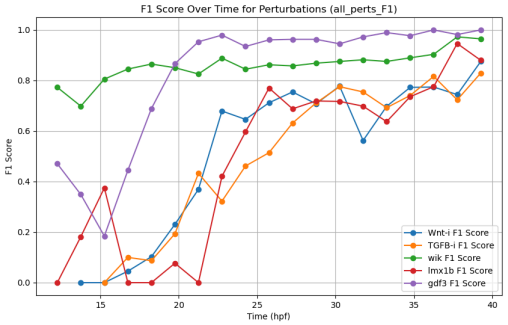
Model 74

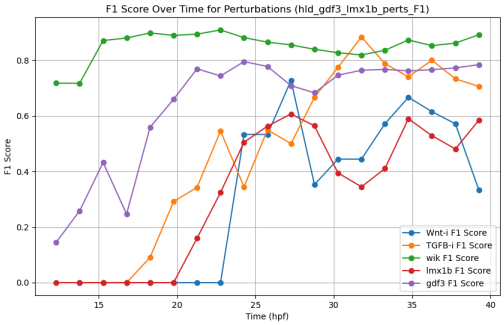




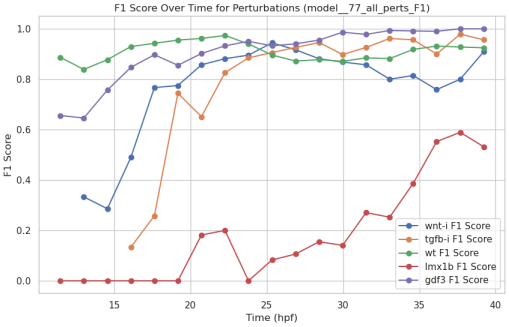
comparing 74 to previous model we were using

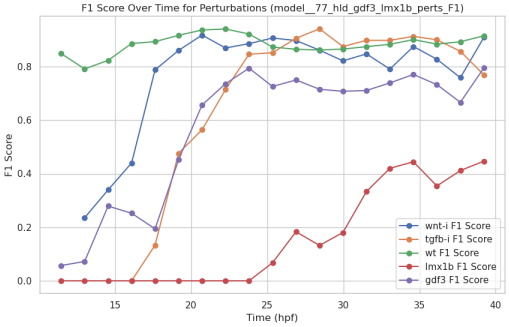
- First striking takeway is that model 74 does a much better job of EXCTRACTING relevant information for classification at earlier time points in comparison to previous model. (note that all the best models do a better job not just 74)
  - compare the purple lines (gdf3)
- Prev model:





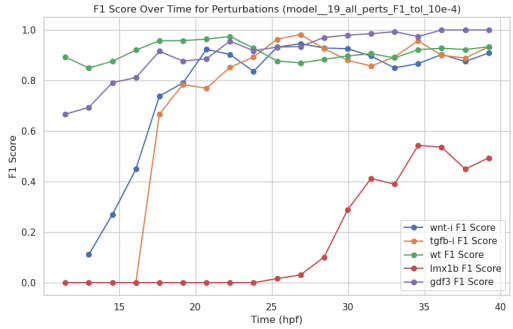
- Model 74:

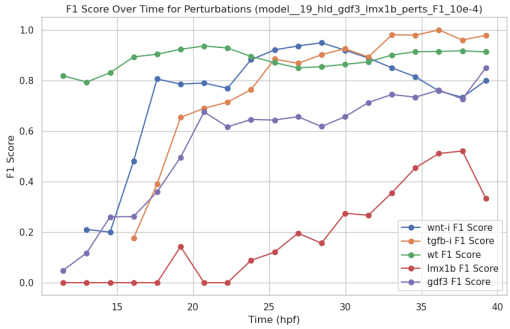




Following up on gains on Classification

- I was curious if increasing the tol (tol: 0.05 --> .10e_4) of classsifcation (letting log reg model run longer) will improve results and the answer is not really by much
  - Increasing the tol of classification:


F1 Score Over Time for Perturbations (model__19_all_perts_F1_tol_10e-4)
F1 Score
Time (hpf)
wnt-i F1 Score
tgfb-i F1 Score
wt F1 Score
lmx1b F1 Score
gdf3 F1 Score


F1 Score Over Time for Perturbations (model__19_hld_gdf3_lmx1b_perts_F1_10e-4)
F1 Score
Time (hpf)
wnt-i F1 Score
tgfb-i F1 Score
wt F1 Score
lmx1b F1 Score
gdf3 F1 Score