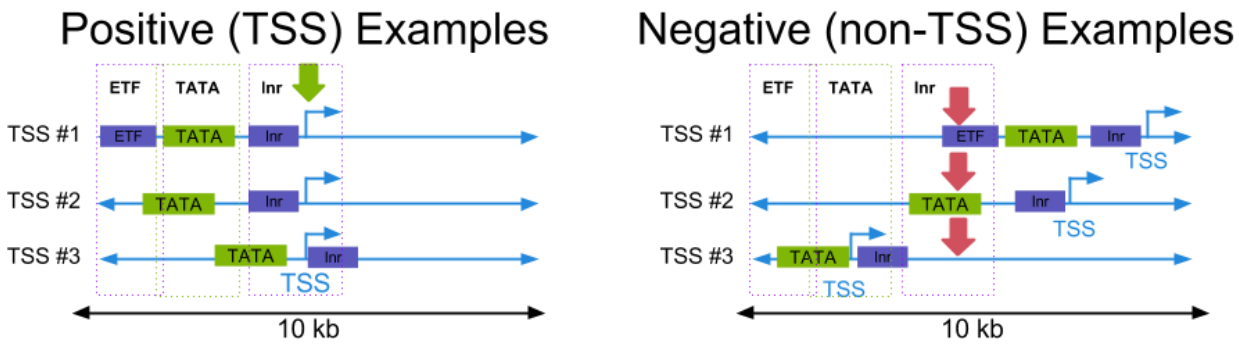
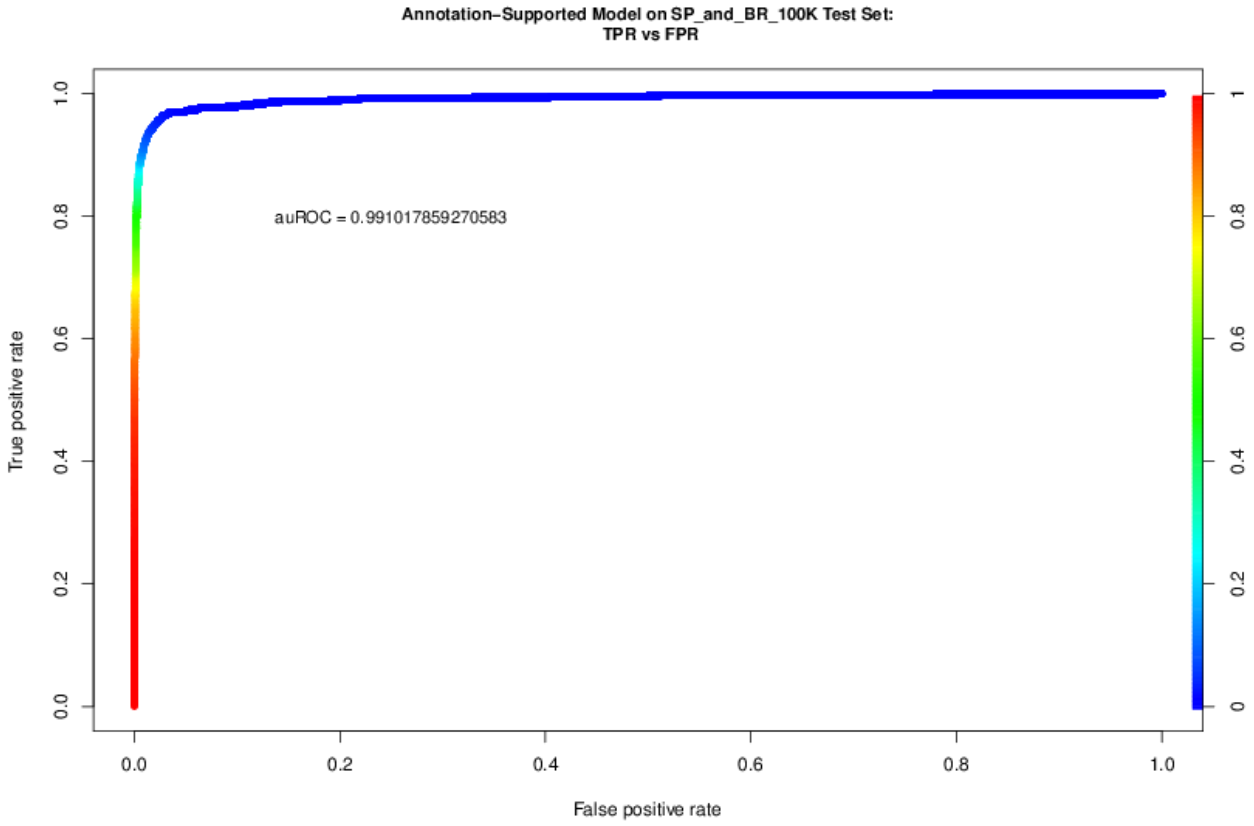
Supplementary Materials

Supplementary Figures

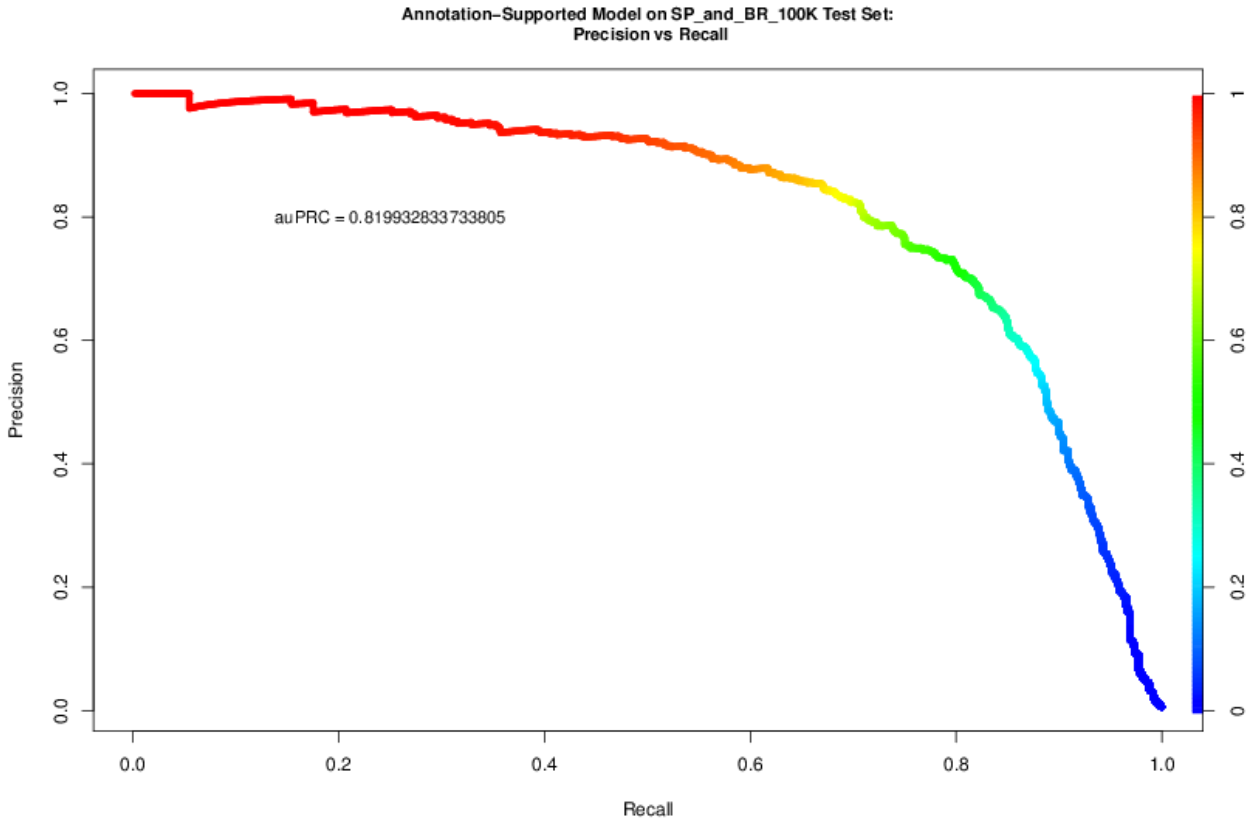


Supplementary Figure 1: Diagram of sequence feature extraction process.

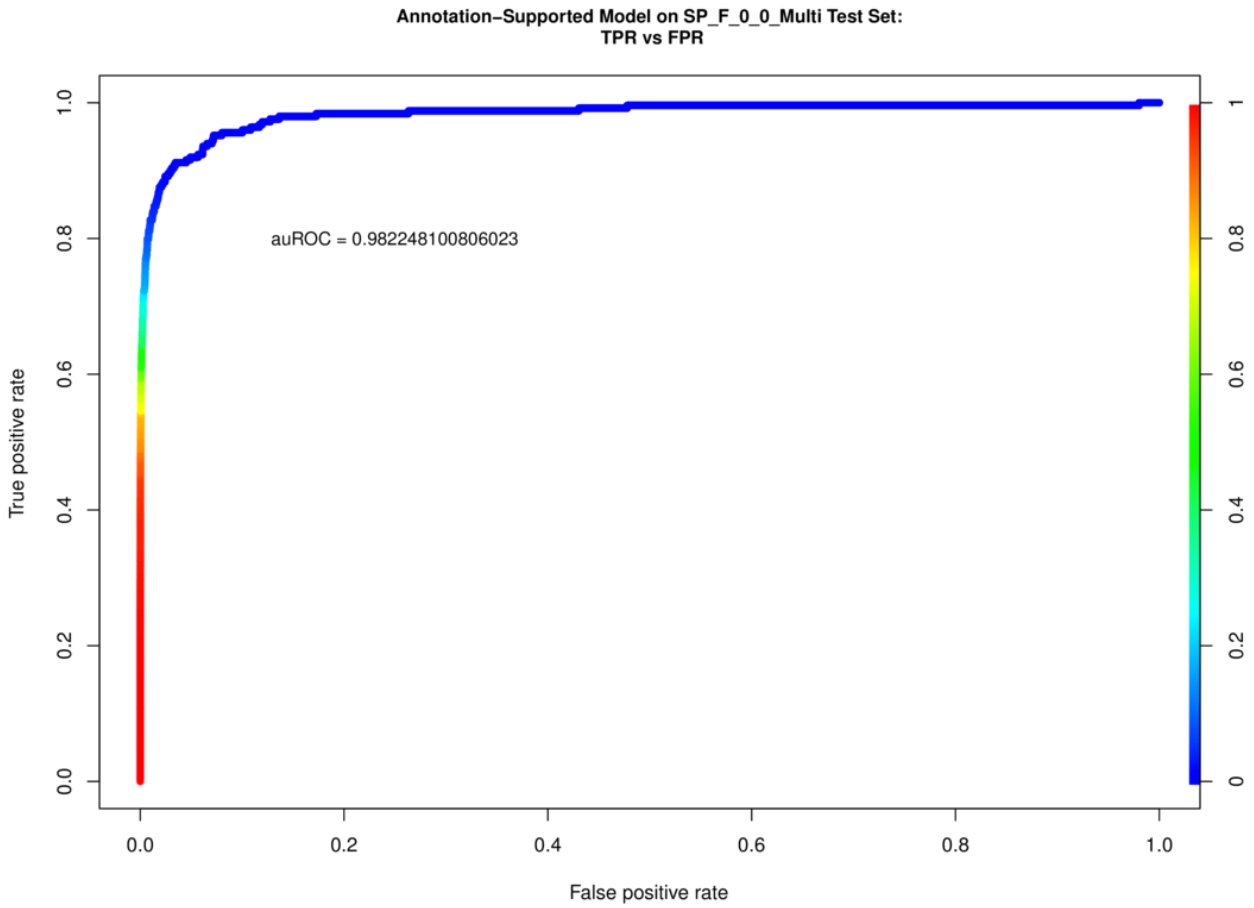
The DNA sequences surrounding experimentally identified TSSs (labeled TSS, green arrow) are extracted and the presence or absence of TFBSs within their ROEs are scored. The ROEs identified for each TF are shown in dotted lines. Red arrows denote the positions of randomly-selected negative examples where no evidence of transcription was supported by the TSS-Seq dataset.



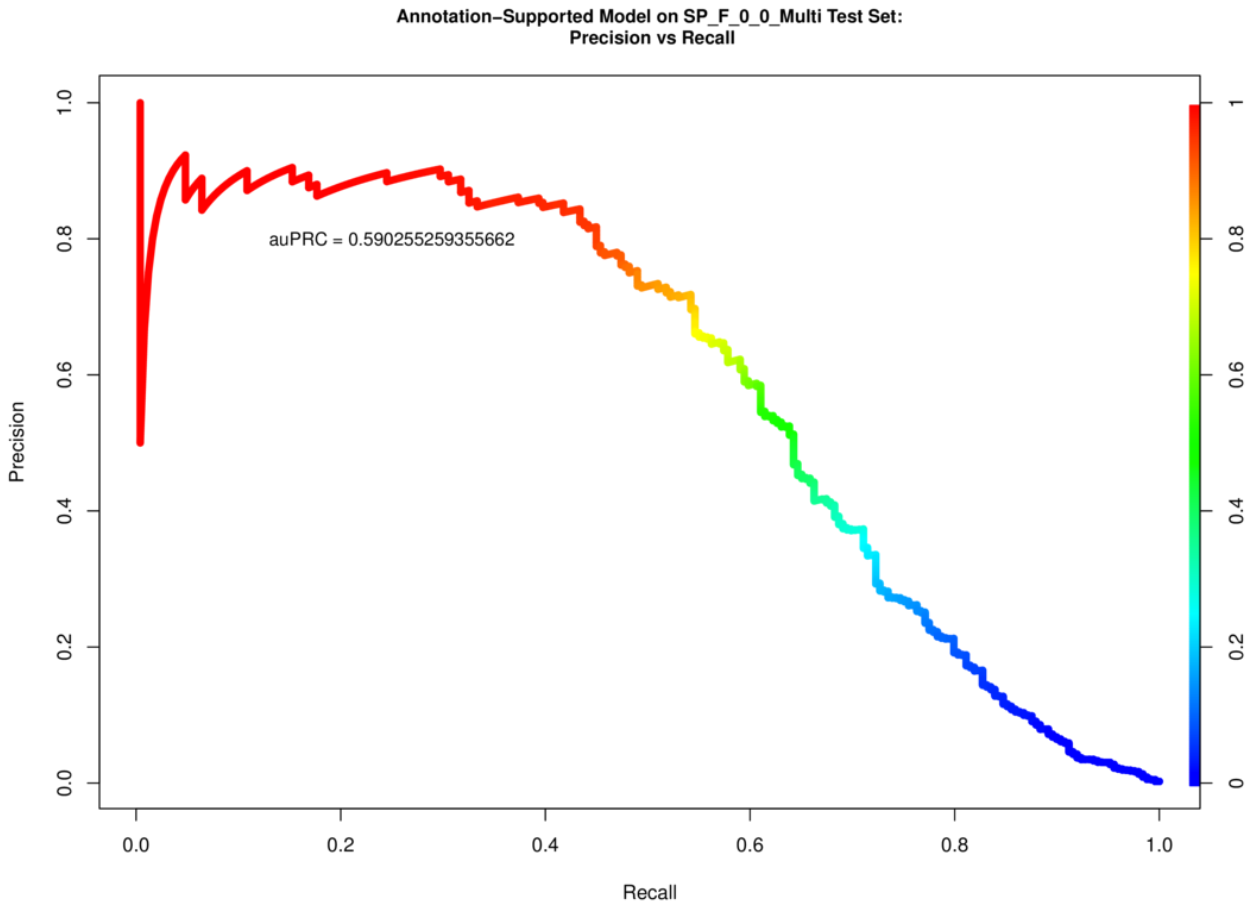
Supplementary Figure 2: ROC Plot of SP + BR (ALL) vs NO Classifier.



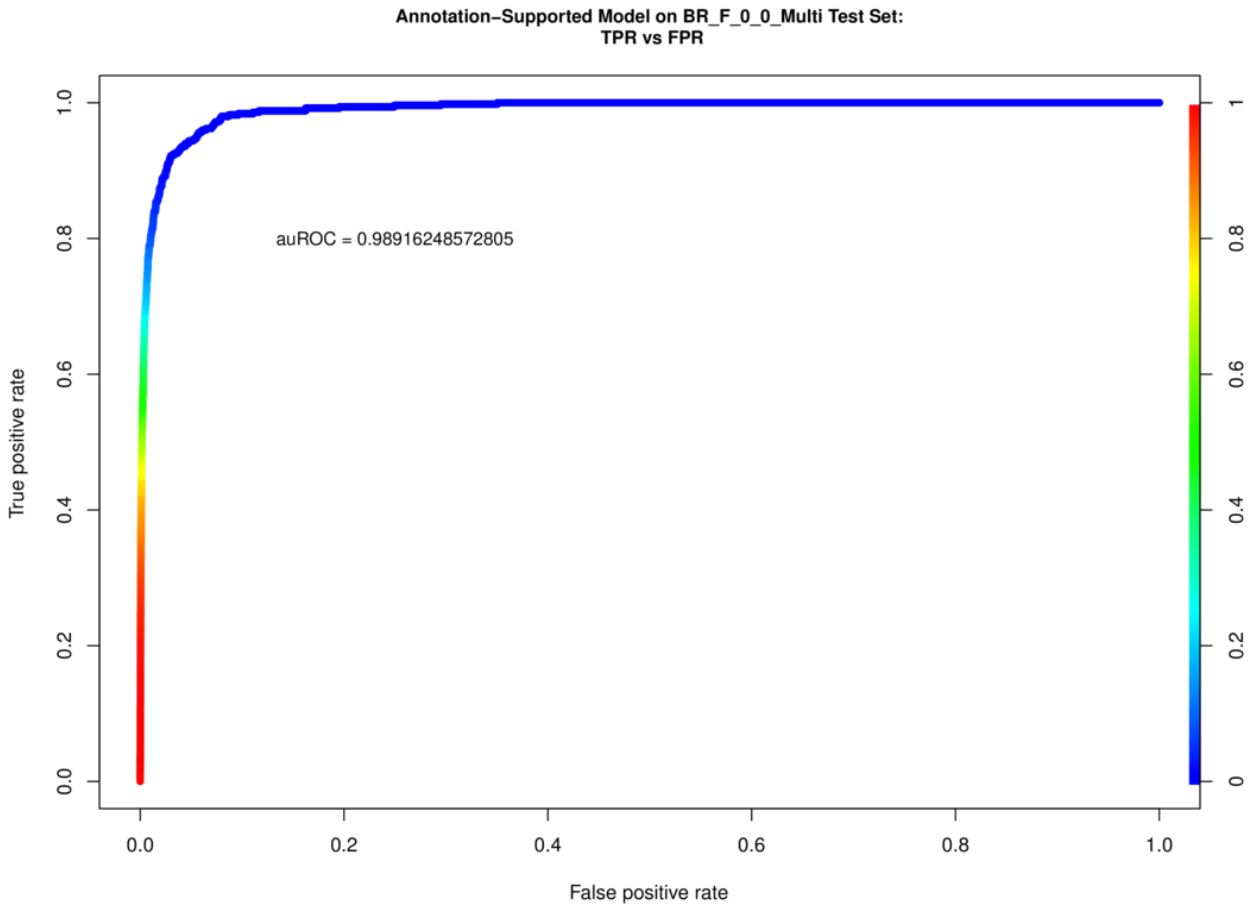
Supplementary Figure 3: PRC Plot of SP + BR (ALL) vs NO Classifier.



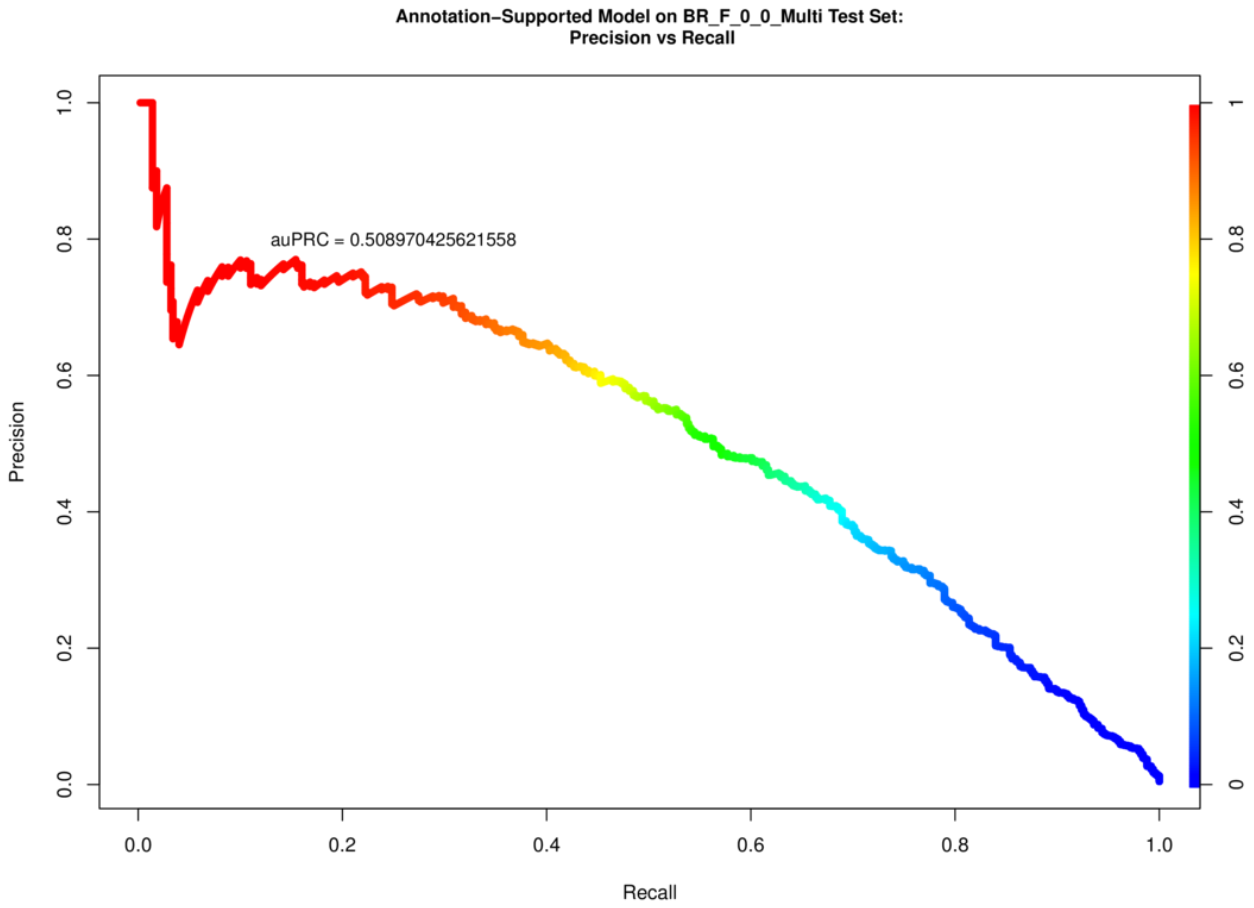
Supplementary Figure 4: ROC plot of the BR vs NoTSS model when used to predict Single Peak initiation patterns.



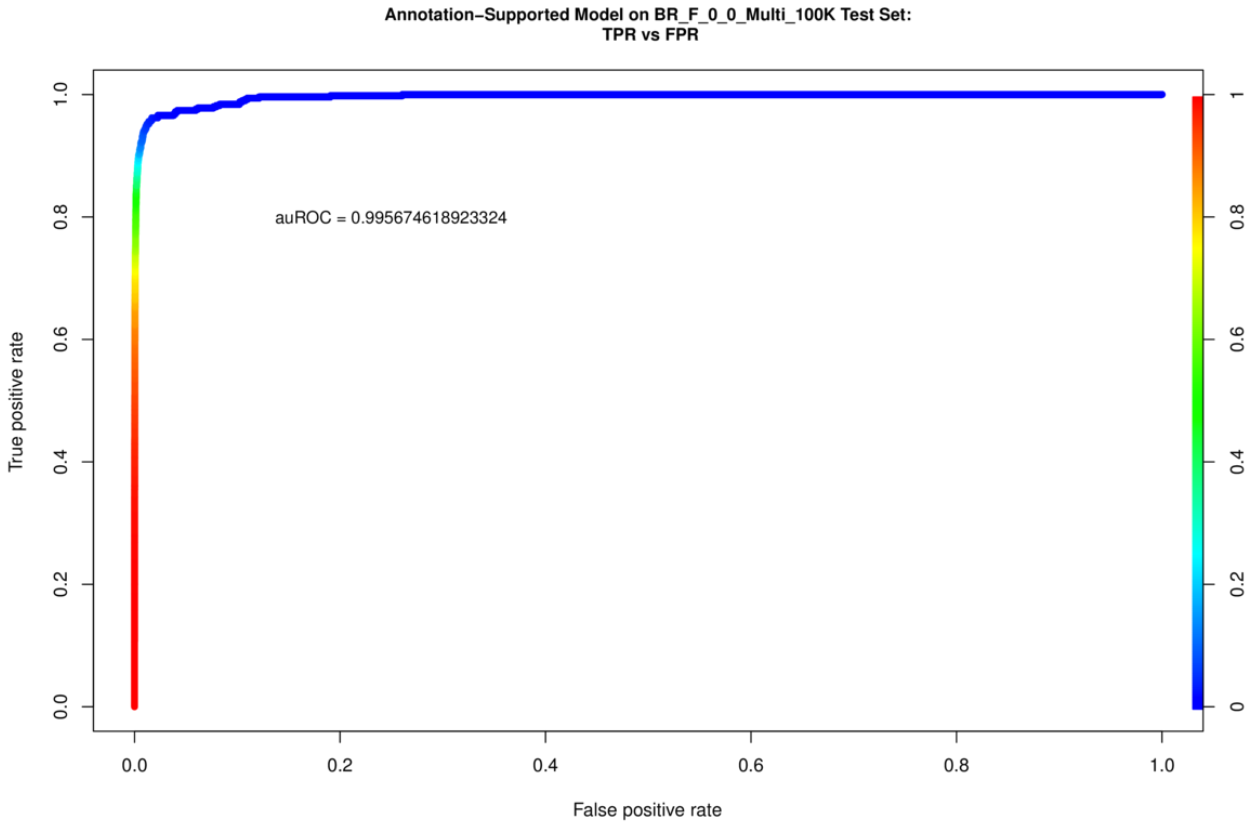
Supplementary Figure 5: PRC plot of the BR vs NoTSS model when used to predict Single Peak initiation patterns.



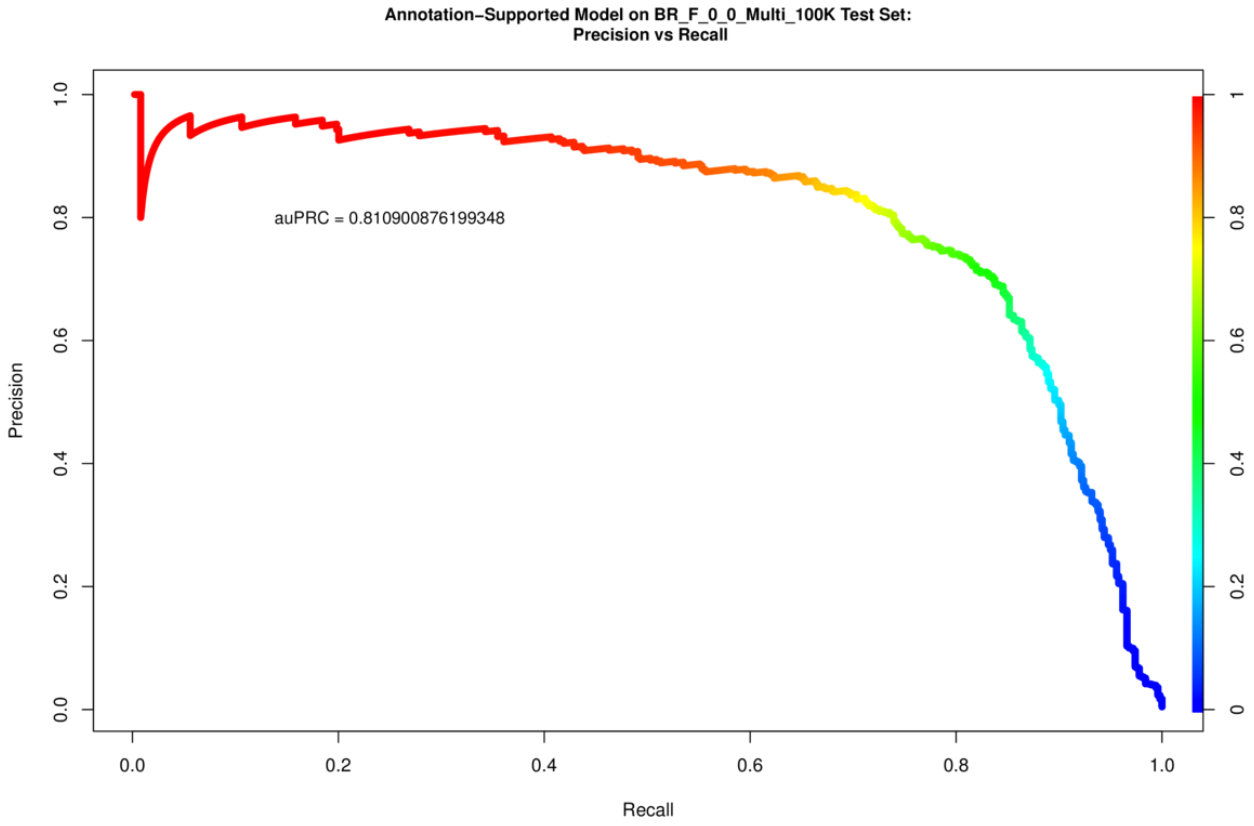
Supplementary Figure 6: ROC plot of the SP vs NoTSS model when used to predict Broad initiation patterns.



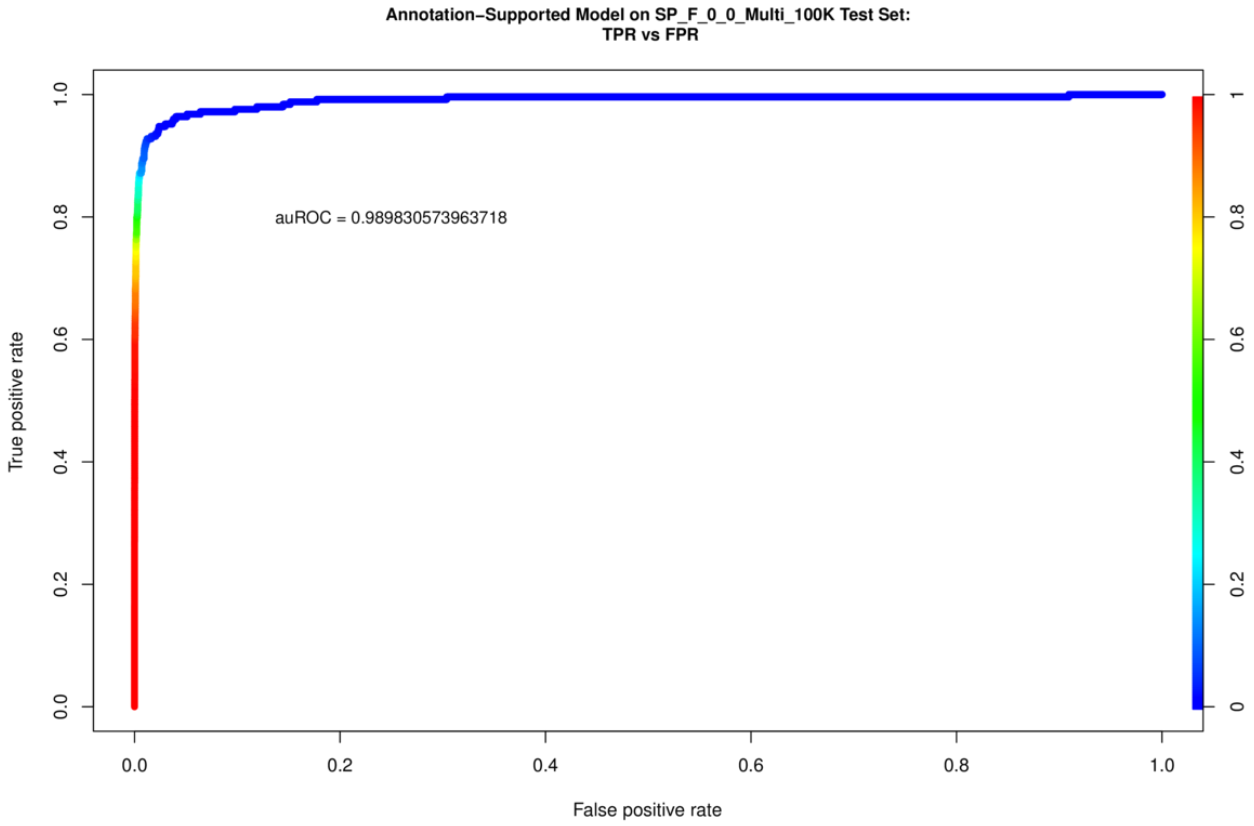
Supplementary Figure 7: PRC plot of the SP vs NoTSS model when used to predict Broad initiation patterns.



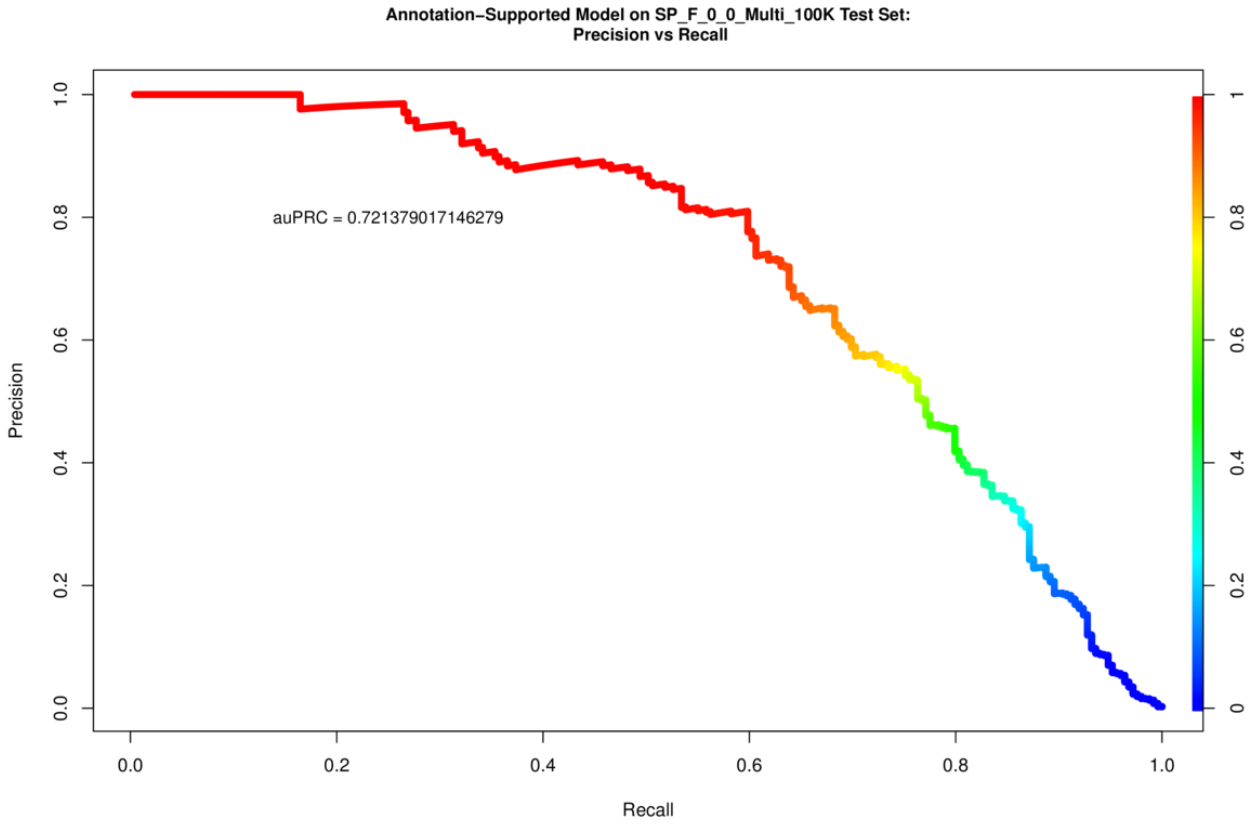
Supplementary Figure 8: ROC of BR vs NoTSS Model.



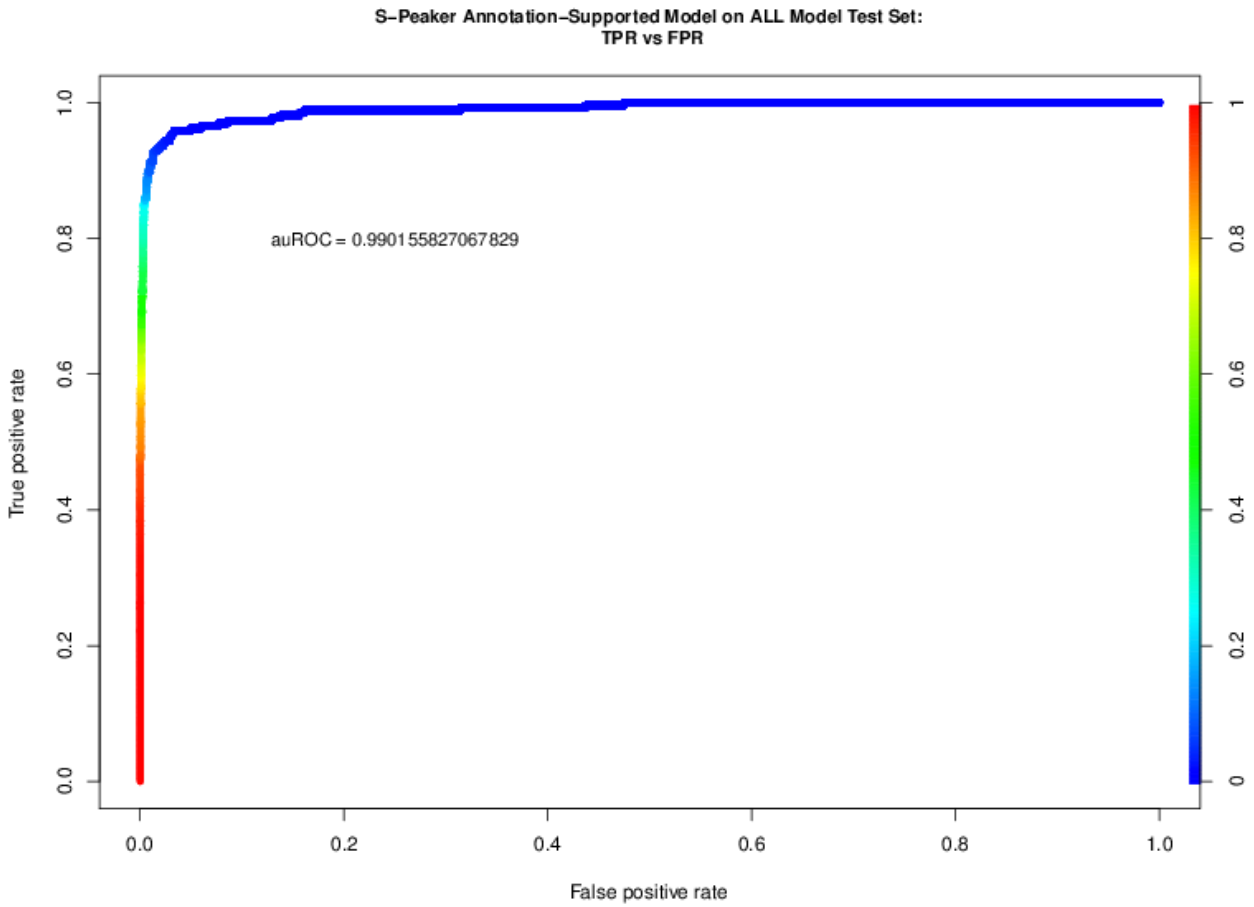
Supplementary Figure 9: PRC of BR vs NoTSS Model



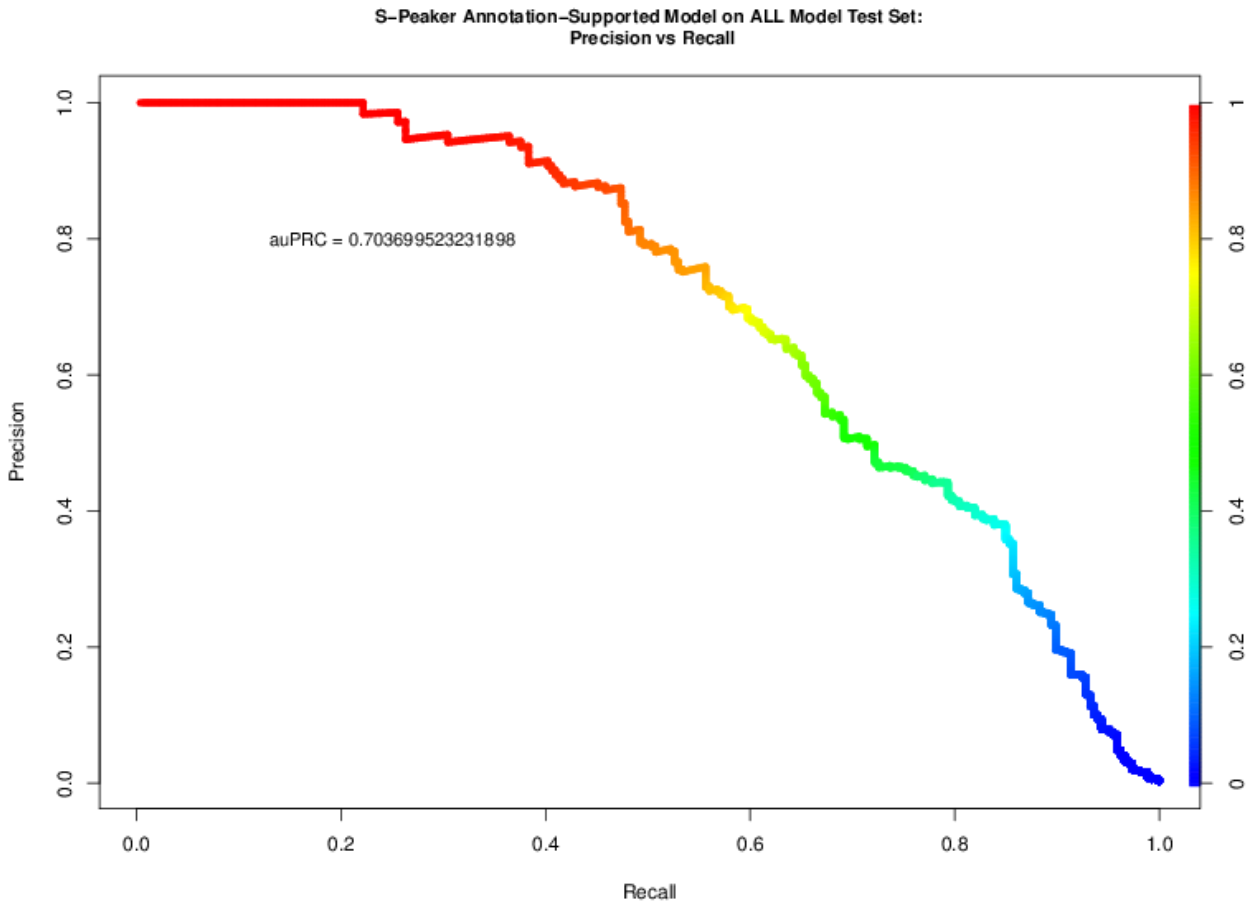
Supplementary Figure 10: ROC plot of SP vs NoTSS Model



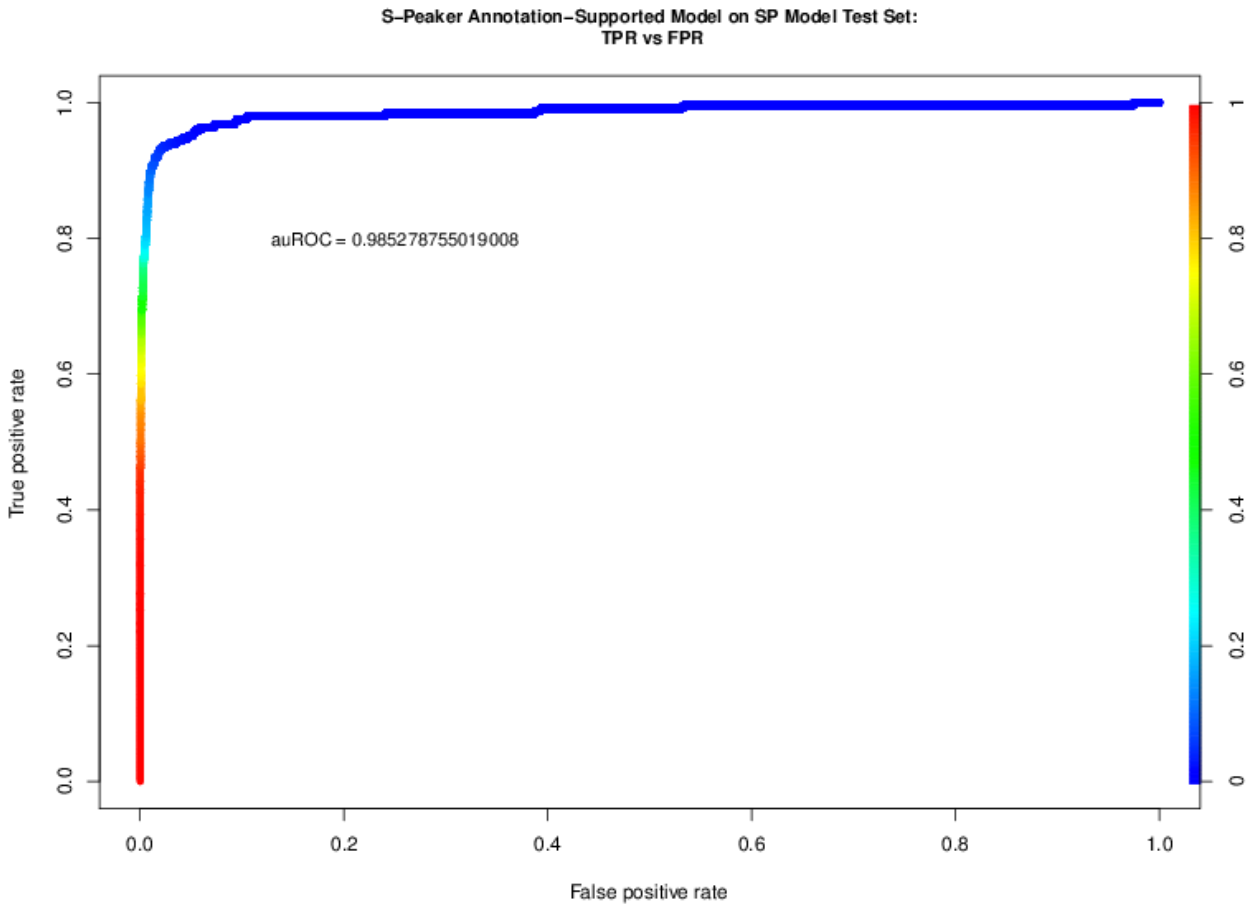
Supplementary Figure 11: PRC plot of SP vs NoTSS Model



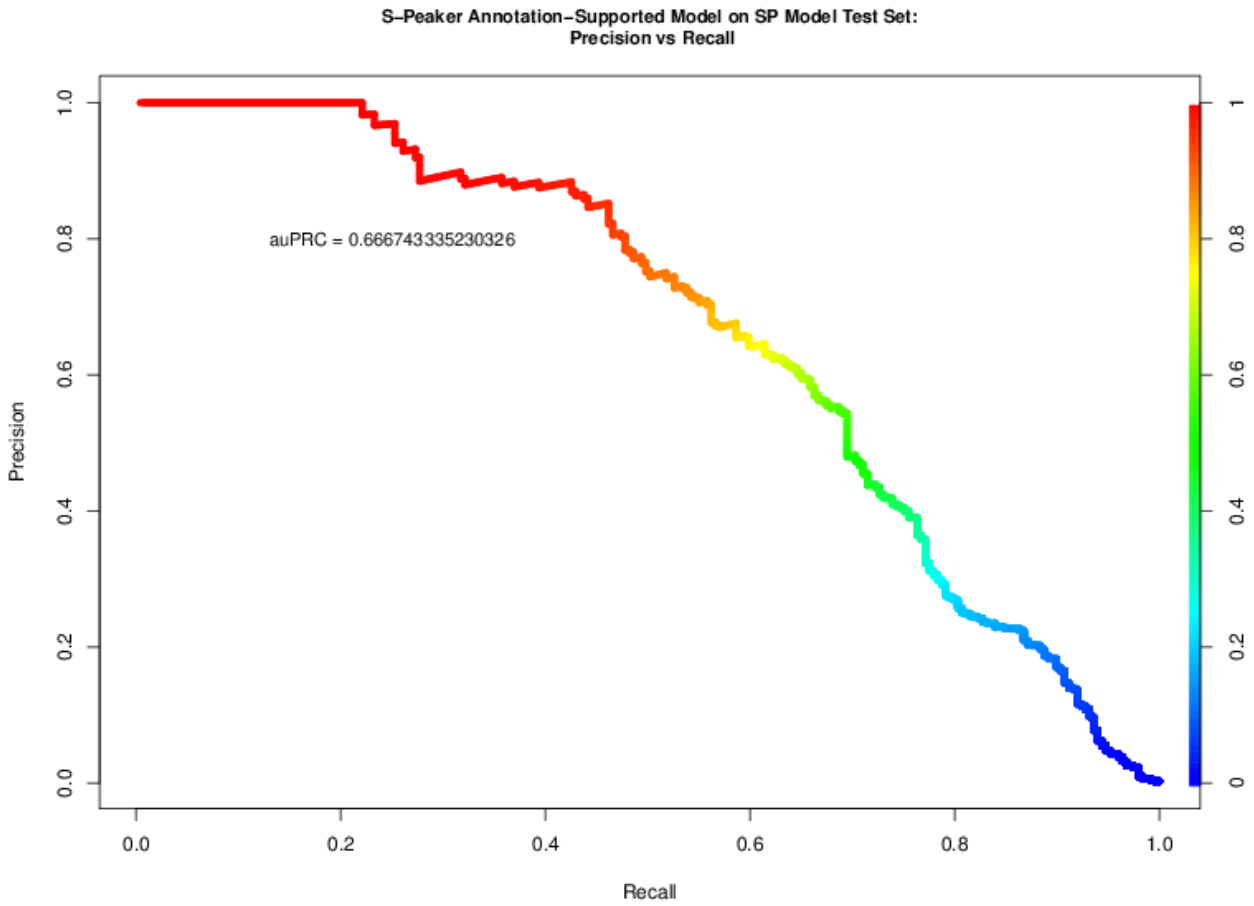
Supplementary Figure 12: ROC plot of S-Peaker Model on ALL Dataset



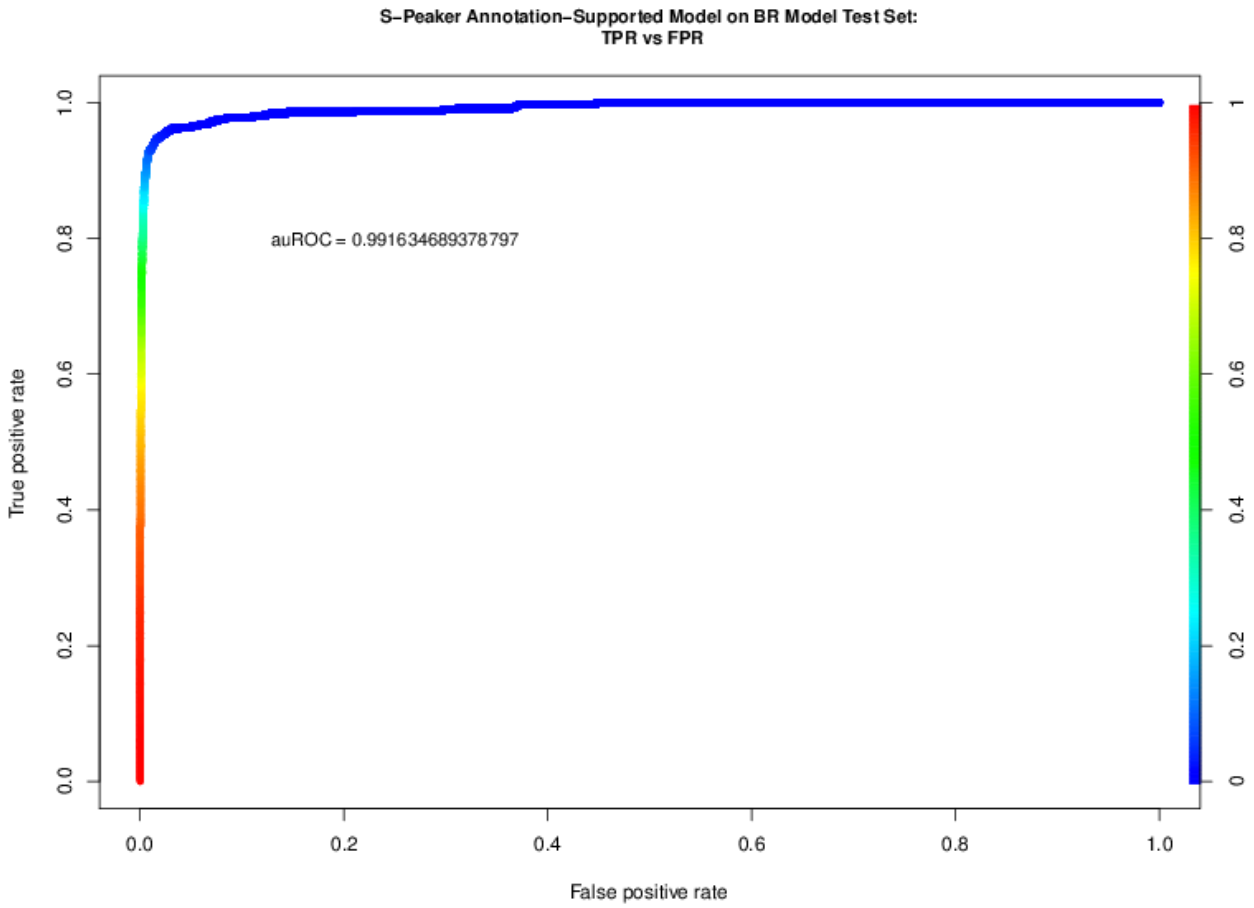
Supplementary Figure 13: PRC plot of S-Peaker Model on ALL Dataset



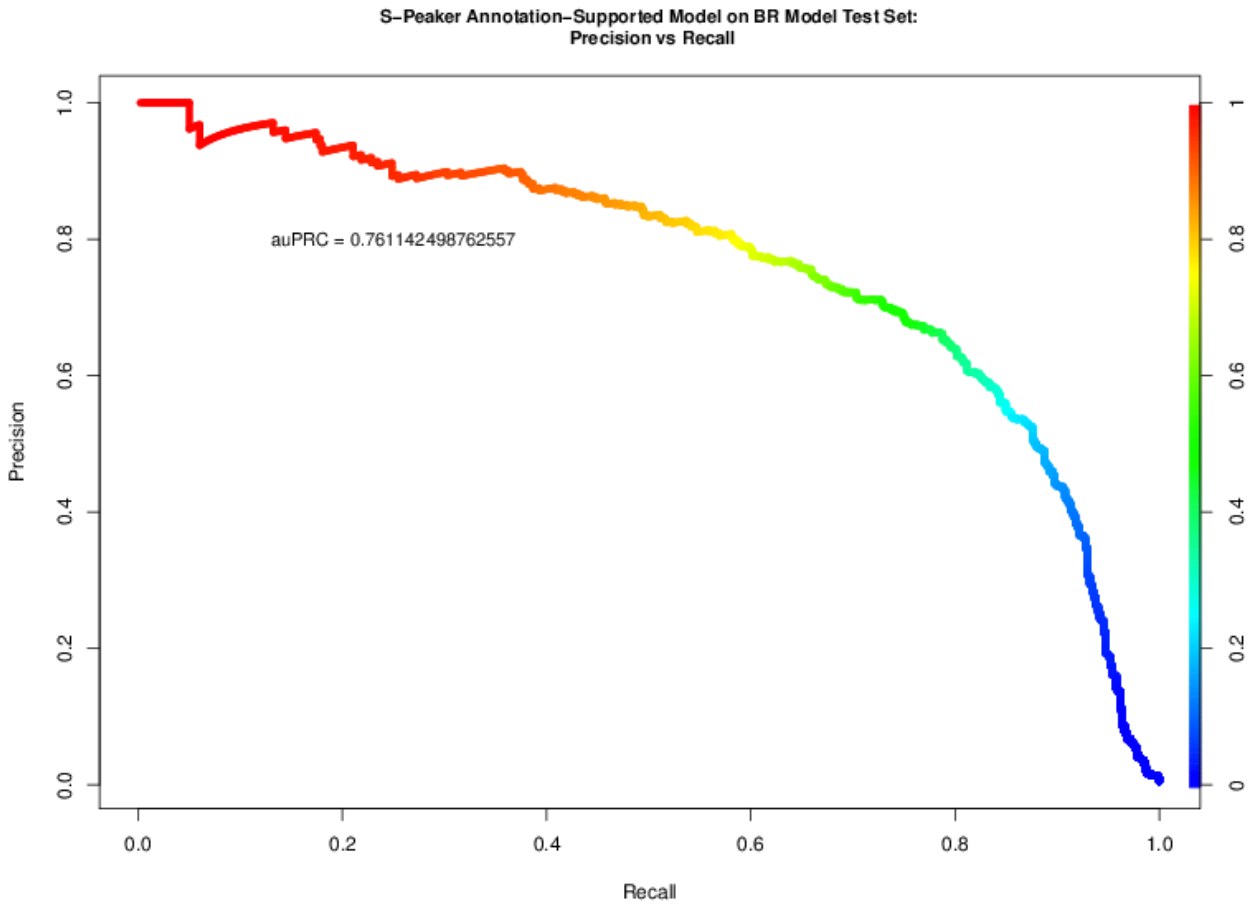
Supplementary Figure 14: ROC plot of S-Peaker Model on SP Dataset



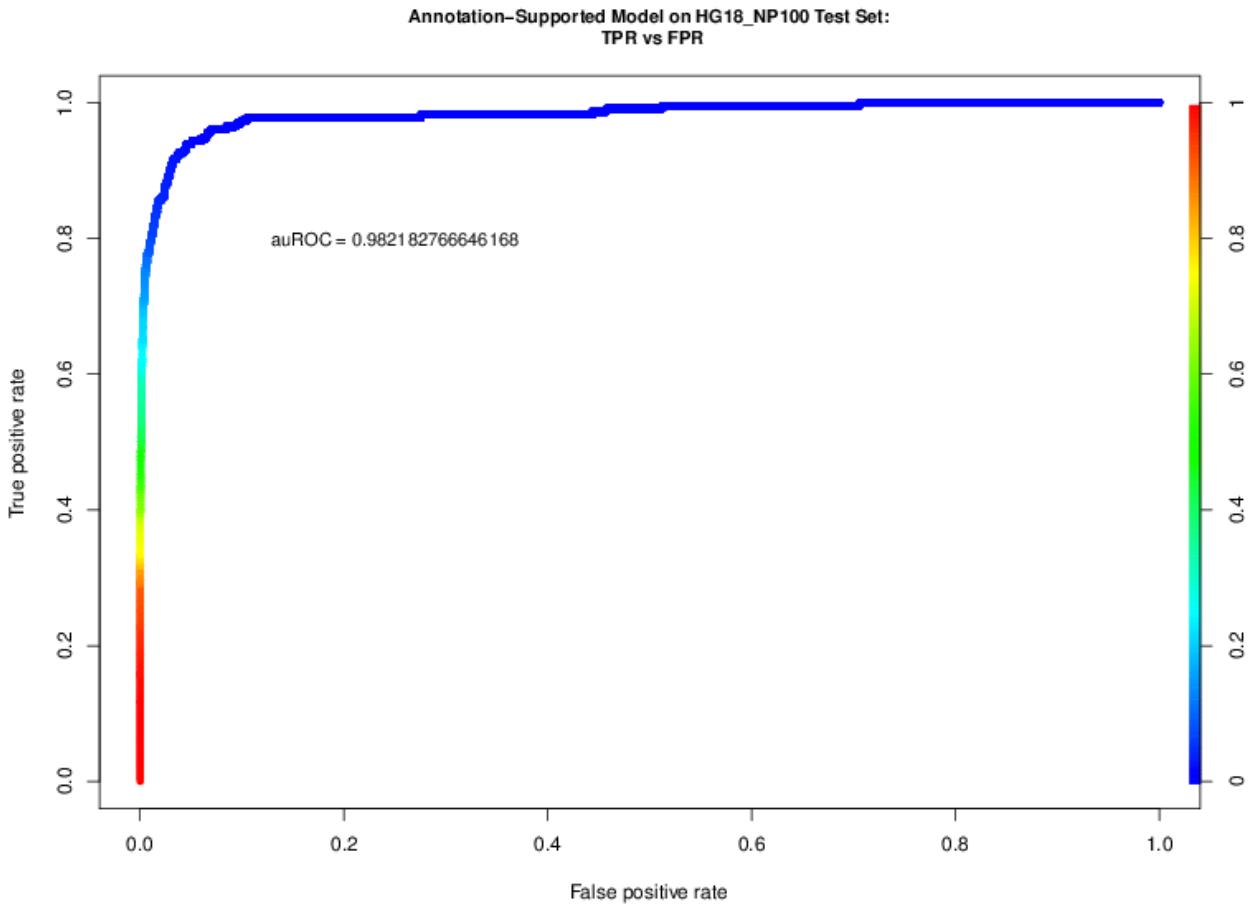
Supplementary Figure 15: PRC plot of S-Peaker Model on SP Dataset



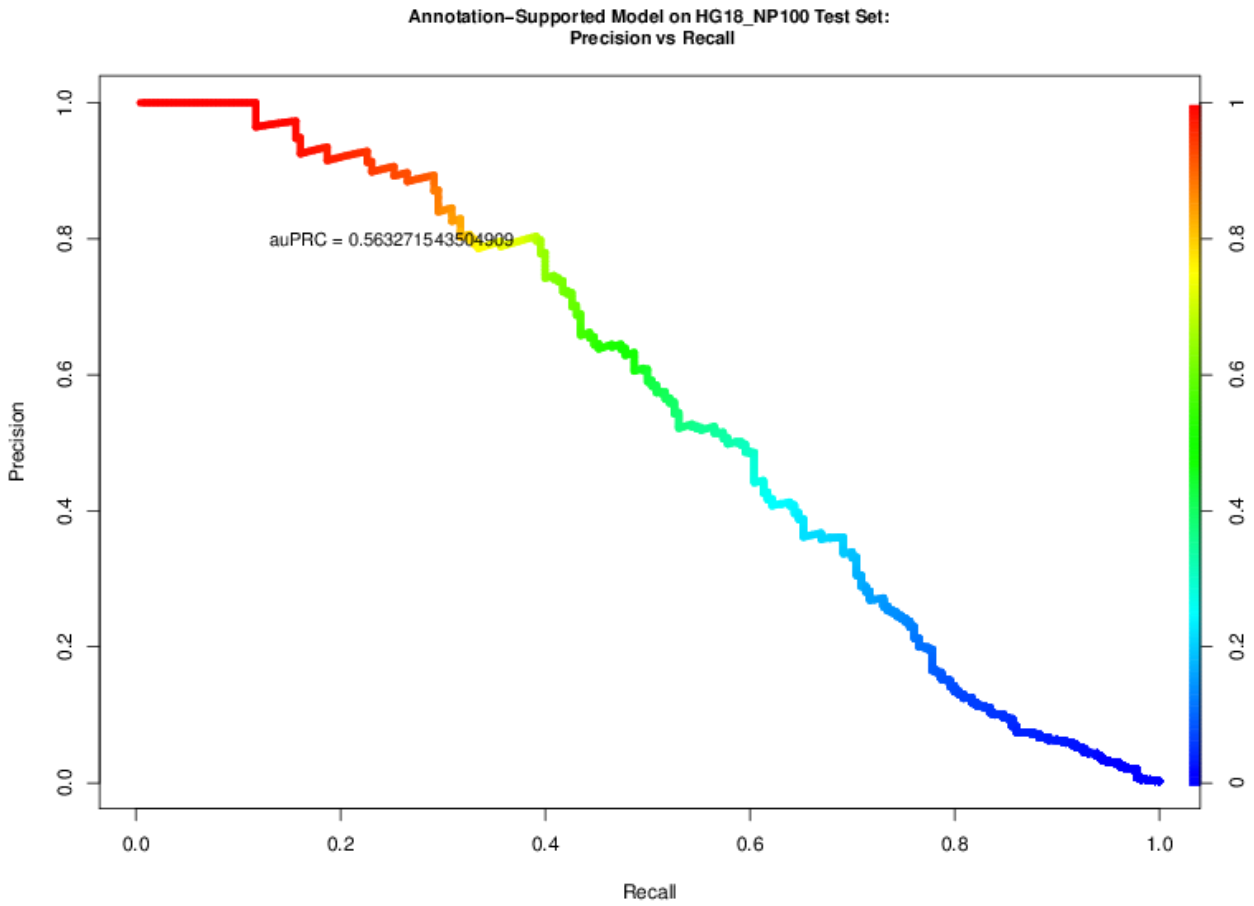
Supplementary Figure 16: ROC plot of S-Peaker Model on BR Dataset



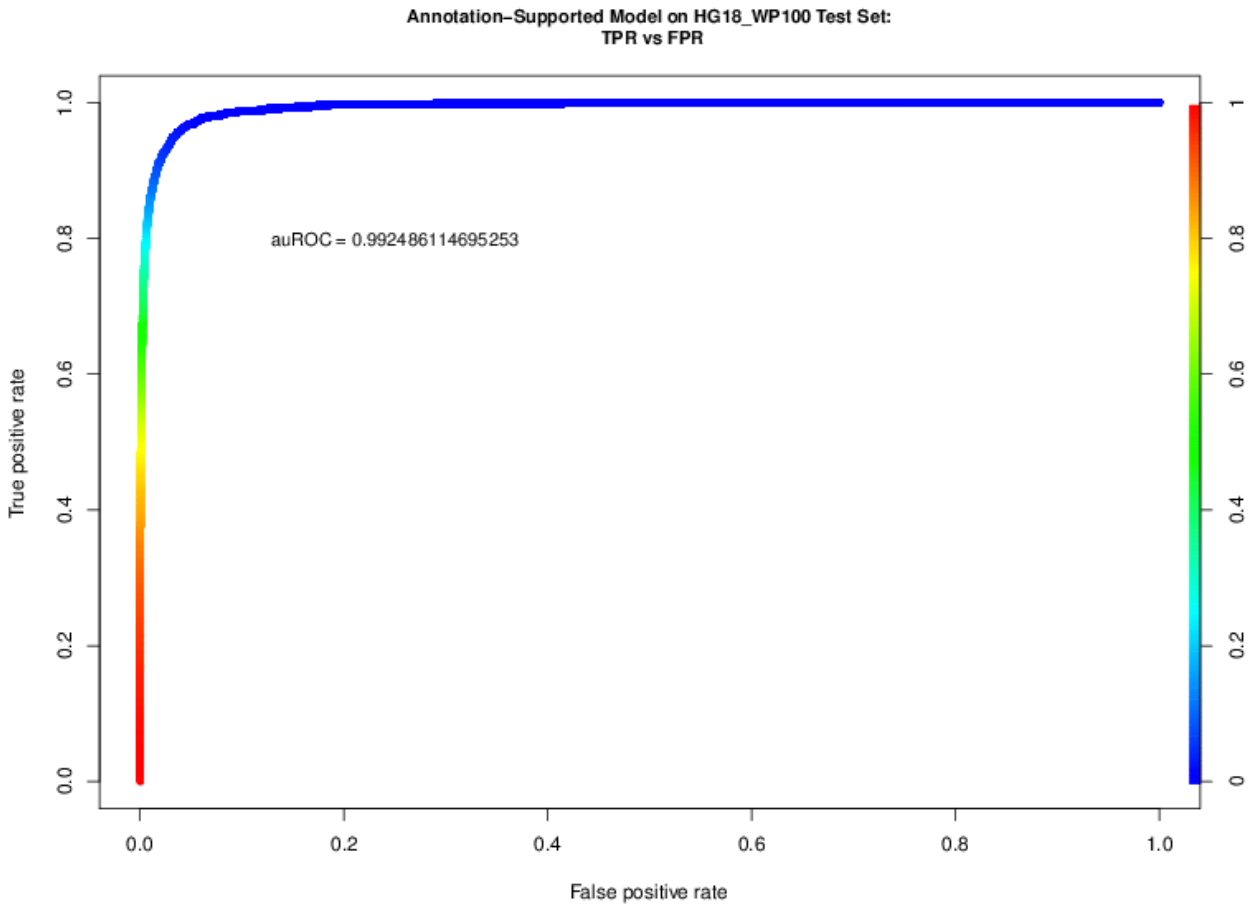
Supplementary Figure 17: PRC plot of S-Peaker Model on BR Dataset



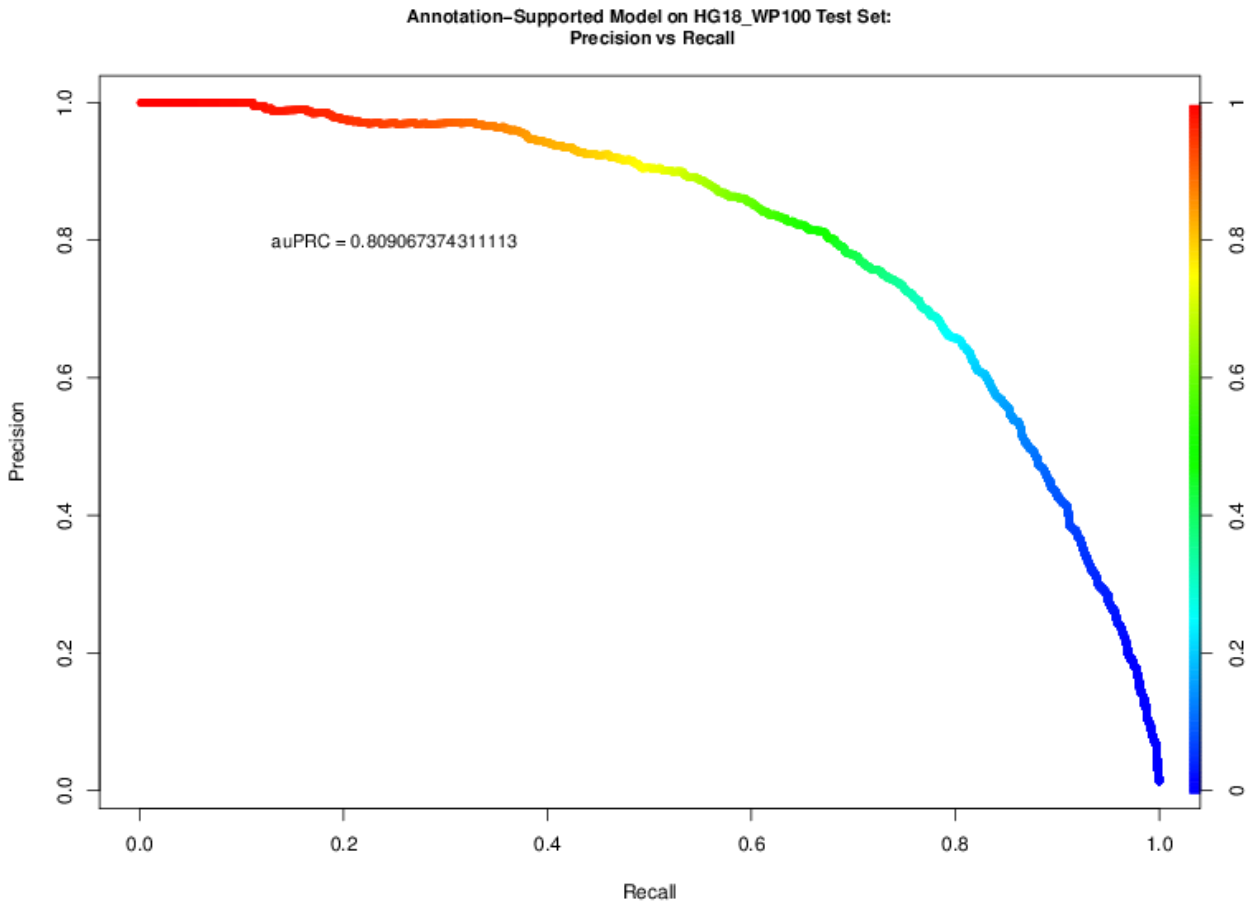
Supplementary Figure 18: ROC Plot of TIPR trained and tested on HG18 Narrow Peak dataset



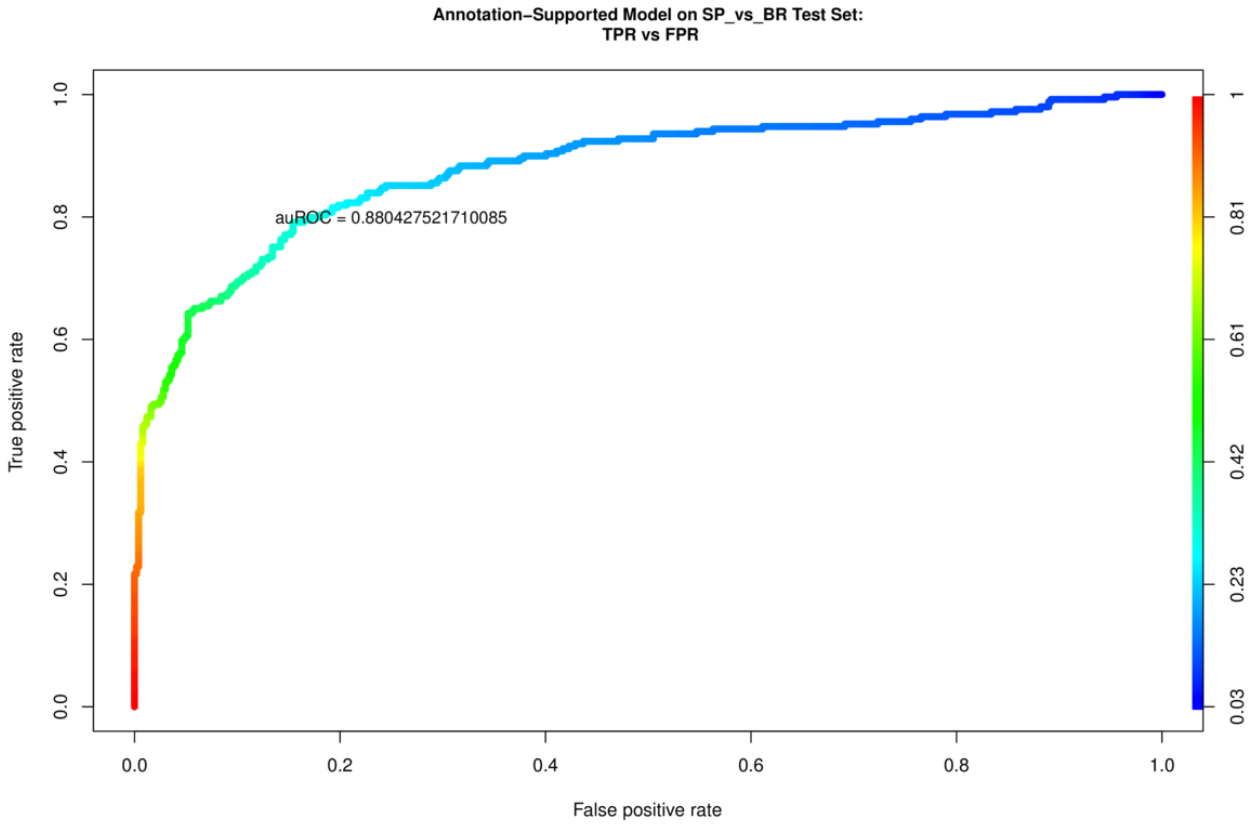
Supplementary Figure 19: PRC Plot of TIPR trained and tested on HG18 Narrow Peak dataset



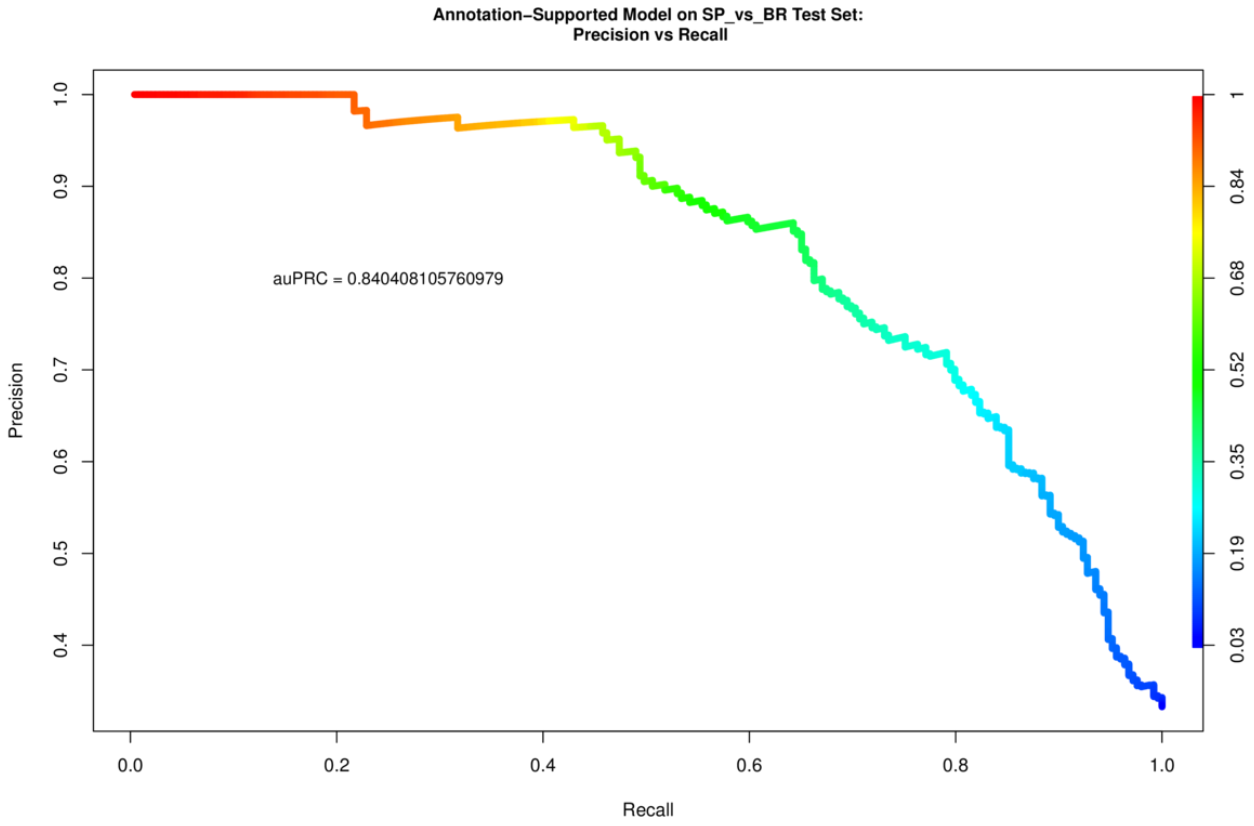
Supplementary Figure 20: ROC Plot of TIPR trained and tested on HG18 Weak (Broad) Peak dataset



Supplementary Figure 21: PRC Plot of TIPR trained and tested on HG18 Weak (Broad) Peak dataset



Supplementary Figure 22: ROC plot of SP vs BR model



Supplementary Figure 23: PRC plot of SP vs BR model

Supplementary Tables

Supplementary tables are provided in the following Excel spreadsheet files.

Supplementary Table 1: Regions of Enrichment defined for each set of TSSs by initiation pattern and DNA strand. The SP and BR sheets are defined on training examples of a single initiation pattern, and were used to construct the SP and BR models. The sheets labeled "Union of SP+BR" were used to build the 3 models used by the MSC classifier. The sheets labeled "ALL" contain ROEs defined by combining both the SP and BR TSSs together. (Supplementary Table 1 – All_Model_ROEs.xls)

Supplementary Table 2: Performance of classifiers in cross-validation tested on the validation partition of each cross-validation fold. This table shows the auROC and auPRC of each model used in the MSC and ALL classifiers built for each cross-validation fold on the validation partition. (Supplementary Table 2 - Cross-Validation Scores.xlsx).

Supplementary Table 3: The feature coefficients of each final MSC sub-model. (Supplementary Table 3 – Feature Weights.xls)

Supplementary Table 4: Confusion Matrices and additional statistics calculated from the test sets of each model. (Supplementary Table 4 – Confusion Matrices.xls)

Supplementary Methods

ROE Identification

The identification of Regions of Enrichment is the first step in the TIPR model training procedure. After the DNA sequence surrounding TSS-Seq-supported TSSs is extracted, each TRANSFAC TFBS PWM is scanned along each extracted sequence, and a log-likelihood score is calculated at each nucleotide. This score is the log-likelihood that the sub-sequence beginning at this nucleotide is drawn from the TFBS PWM distribution compared to the promoter background distribution. This score is equivalent to the (log of) the quantity P(TFBS)/P(Background). The promoter background is calculated using a first-order Markov model over 500 nt surrounding the TSS (250 nt upstream and 250 nt downstream of the TSS). After this procedure is performed over the sequence surrounding each TSS, positive scores are averaged across all TSSs, producing an average log-likelihood score at each nucleotide, where position zero is the mode of each TSS tag cluster identified through TSS-Seq.

Following this scanning procedure, ROEs are identified by first locating the nucleotide with the maximum averaged log-likelihood score within 100 nt of the TSS (upstream or downstream). Starting from this location, the ROE is expanded upstream and downstream until the average log-likelihood score falls below the overall average (average of all average log-likelihood scores at all nucleotides surrounding the TSS within 2 kb of the TSS) for at least 5 nt. These positions define the boundaries of the TFBS's Region of Enrichment. These regions are further subdivided into 5 overlapping windows of equal size, plus an additional 2 flanking the upstream and downstream edges (of the same width). The 7 windows are each considered separately during feature extraction. This entire procedure is performed on both the forward and reverse strands of

DNA, identifying ROEs present on both strands. Further details are provided in Megraw *et al.* (2009), a visualizations are available in Supplementary Figure 2 and Megraw *et al.* (2009), Figure 3.

Sequence Feature Extraction

After regions of enrichment have been identified, the DNA sequence surrounding TSSs are transformed into numerical features characterizing the presence of TFBSs in the sequence which falls within the TF's ROE. The log-likelihood procedure described above is repeated, except that only sequences falling within a specific TFBS's ROE are considered. To generate features for a single genomic location (either as a positive training example containing a TSS, a negative non-TSS training example, or to predict the probability of transcription initiating at that nucleotide), the sequence surrounding the nucleotide is extracted and the genomic positions of ROEs relative to the genomic position under investigation are calculated (Supplementary Figure 3).

Within the ROE of each TFBS, log-likelihood scores are computed as described in the ROE Identification section. A numerical score is produced for each ROE sub-window by summing all positives scores of all nucleotides which fall within the window, producing a total of 7 features for each TFBS per strand. An additional 3 features containing dinucleotide sequence enrichments are included with each example. Sequence enrichment features are simply the proportion of bases within 250 nt of the genomic location which contain either of the nucleotides in the dinucleotide set.

The TIPR-TFBS-Scan program performs this scoring function on input DNA sequence as FASTA files and produces as output a text or binary file containing the numerical features of each input sequence. The binary format is used in this model to decrease file size and increase training efficiency.

l1_logreg modifications

The l1_logreg software package (Koh *et al.*, 2007) is a software package for efficiently training L1-regularized logistic regression models. We have modified l1_logreg to support the binary files produced by the TIPR-TFBS-Scan application. This increases training and testing speed by reducing the time required to load large input files containing thousands of features and examples. The modified l1_logreg package is included in supplementary materials.

Model Training and Selection of Model Parameters

Models are trained using 80% of available data, with the remaining 20% used for testing. Training data is used for both ROE identification and model training, while testing data is never used to inform the model in any way. Data was randomly partitioned into training and testing sets. Model parameters are selected using 10-fold cross-validation as follows:

1. 80% of training data is used for training partitions, 10% for selection of parameter λ, and 10% for selection of parameter *d*.
2. The l1_logreg package is used to compute an approximation of the regularization path at 23 points, with a minimum value of λ=0.0001 on the training partition of each fold.

3. After the regularization path has been computed, the AUROC of at each point is computed on the validation partition of each fold. The λ yielding the highest AUROC of each fold is recorded.
4. The model with optimal λ of each fold found in step 3 is used to classify the examples in the remaining 10% used for selection of parameter *d*. Using these labeled examples, probability threshold values of *d* between 0.0 and 1.0 (in increments of 0.02) are used to predict the class label. For each value of *d*, the F1 score of the resulting classification is calculated, and the *d* which maximizes this score is recorded. The average of these maximum values of *d* is computed across all cross-validation folds and used as the *d* value for the final model. This value represents the probability threshold which will result in the optimal F1 score when used to predict class labels from the probability output of a binary classifier prediction.
5. The optimal λ values computed in step 3 are averaged together to choose the λ penalty of each final model. Final models are trained using these averaged λ values on the entire training set.

Multi-Class Prediction Models

During the development of TIPR, we experimented with several multi-class prediction algorithms. We describe the details of three models here, with results given in the next section. All models are similar, but vary in the order in which the three sub-models (Table 1 in the manuscript) are applied.

Model 1: In this model, the ALL model (trained using a combination of SP and BR initiation patterns) is used to determine the probability of transcription initiation a particular genomic location under investigation. If the ALL model predicts the location is transcribed (probability of initiation greater than or equal to the ALL model's *d* parameter), the SP vs BR model is used to predict if the genomic region under investigation is likely to form an SP or BR initiation pattern.

Model 2: This model uses both of the initiation pattern-specific TSS prediction sub-models (SP vs NO and BR vs NO) to predict if a genomic location is transcribed by either model. The following algorithm is applied. P(x) represents the probability output of the specified model, while d(x) represents the value of the *d* parameter for a specific model.

```
If P(SP vs NO) > d(SP vs NO):  // SP vs NO model predicts site is transcribed
        if P(SP vs BR) > d(SP vs BR):  // SP vs BR model predicts site is an SP
                predict SP
        else: predict NO
Else If P(BR vs NO) > d(BR vs NO): // BR vs NO model predicts site is transcribed
        If P(SP vs BR) < d(SP vs BR):  // SP vs BR model predicts site is an SP
                predict BR
        else: predict NO
Else: predict NO
```

Model 3: This is the MSC model reported on and described in the manuscript. This is similar to Model 2, but the predicted label of the TSS (SP vs NO/BR vs NO) and initiation pattern (SP vs BR) models are not required to agree. The following algorithm is applied:

```
If P(SP vs NO) > d(SP vs NO) or P(BR vs NO) > d(BR vs NO):
        if P(SP vs BR) > d(SP vs BR): predict SP
        else: predict BR
else: predict NO
```

Construction of FANTOM4 Human TSS Prediction Model

To compare TIPR to the chromatin-based TSS prediction model described in Rach *et al*., 2011, we used the TSS location and initiation pattern classification dataset provided in the supplementary materials of Rach *et al*., 2011 (https://ohlerlab.mdc-berlin.de/publications/29/).

As in the TIPR model, the dataset was partitioned into training and testing sets, with 80% of the TSSs being used for training. 100,000 random locations from the hg18 genome, along with randomized locations drawn from annotated exons, were used to construct the negative set. Unlike in the TIPR model, negative examples drawn from regions surrounding TSSs were not explicitly filtered to ensure that there were no TSSs within the regions used to draw negatives. This is the same procedure that was used to generate the negative set in Rach *et al.*, 2011. Our algorithm for selecting regions of the genome which showed no evidence of transcription (negative examples) could not be applied to the dataset used to construct the Rach *et al*., model. This is likely because our algorithm finds only large swaths of the genome (4 kb) which show no evidence of transcription in regions surrounding TSSs. The dataset used in the Rach *et al.*, model likely has few regions where no transcription occurs. A better negative filtering algorithm which identified smaller regions (10—100 nt) with no transcription could easily be applied, and would likely lead to increased performance of the TIPR model.

Supplementary Results: Multi-Class Prediction Results

The results of the MSC model are reported in the main manuscript. In this section, we compare the 3 multi-class models described in Supplementary Methods: Multi-Class Prediction Models. Overall, the 3 models performed approximately the same when evaluated on several metrics, though Model 3 identified the most SP and BR true positives correctly. In these results, class 0 is a negative (non-TSS) example, class 1 is an SP TSS, and class 2 is a BR TSS.

Note that because these results were collected during development of the models, these results compare the 3 models with a testing set containing using only approximately 15,000 negative examples instead of the 100,000 reported in the manuscript.

Model 1:

```
Confusion Matrix and Statistics

          Reference
Prediction    -1     1     2
        -1 15625    49    84
        1     43   156    55
        2     40    44   360

                     Class: -1 Class: 1 Class: 2
Sensitivity             0.9947  0.62651  0.72144
Specificity             0.8222  0.99395  0.99474
Pos Pred Value          0.9916  0.61417  0.81081
Neg Pred Value          0.8811  0.99426  0.99132
Prevalence              0.9545  0.01513  0.03032
Detection Rate          0.9495  0.00948  0.02188
Detection Prevalence    0.9576  0.01544  0.02698
Balanced Accuracy       0.9085  0.81023  0.85809
Micro F1:  0.980858
Macro F1:  0.7923115
```

Model 2:

Confusion Matrix and Statistics

Prediction	Reference -1	1	2
-1	15613	49	79
1	25	156	48
2	70	44	372

	Class: -1	Class: 1	Class: 2
Sensitivity	0.9940	0.62651	0.74549
Specificity	0.8289	0.99550	0.99286
Pos Pred Value	0.9919	0.68122	0.76543
Neg Pred Value	0.8671	0.99427	0.99205
Prevalence	0.9545	0.01513	0.03032
Detection Rate	0.9488	0.00948	0.02261
Detection Prevalence	0.9566	0.01392	0.02953
Balanced Accuracy	0.9114	0.81100	0.86917

Micro F1: 0.980858

Macro F1: 0.8003196

Model 3: MSC Classifier

```
Confusion Matrix and Statistics

          Reference
Prediction    -1     1     2
        -1 15605    45    72
        1     32   160    55
        2     71    44   372

                     Class: -1 Class: 1 Class: 2
Sensitivity             0.9934 0.642570  0.74549
Specificity             0.8436 0.994632  0.99279
Pos Pred Value          0.9926 0.647773  0.76386
Neg Pred Value          0.8597 0.994509  0.99205
Prevalence              0.9545 0.015131  0.03032
Detection Rate          0.9483 0.009723  0.02261
Detection Prevalence    0.9554 0.015010  0.02959
Micro F1:  0.980615
Macro F1:  0.7975752
```