














A Preliminary Comparison of P-Tool Consistency

Javier Murillo¹(✉) , Flavio Spetale¹ , Elizabeth Tapia¹ ,
Flavia Krsticevic⁶ , Olivier Cailloux² , Serge Guillaume³ ,
Gustavo Vazquez⁴ , Tamara Fernandez⁴ , Sebastien Destercke⁵ ,
Sergio Ponce⁶ , and Pilar Bulacio¹ 

¹ CIFASIS-CONICET, Univ. Nacional de Rosario, Rosario, Argentina
murillo@cifasis-conicet.gov.ar

² Université Paris-Dauphine, CNRS, Paris, France

³ ITAP, Irstea, Montpellier SupAgro, Univ. Montpellier, Montpellier, France

⁴ Universidad Católica del Uruguay, Montevideo, Uruguay

⁵ Université de Technologie de Compiègne, Compiègne, France

⁶ Universidad Tecnológica Nacional, Regional San Nicolás, Buenos Aires, Argentina

Abstract. Many Bioinformatics tools, known as p-tools, have been developed to predict the effect of single nucleotide polymorphisms (SNPs) on gene functionality, in an effort to reduce the need for *in-vivo* assays. However, the large number of p-tools available and the heterogeneity of their output make their selection and comparison difficult. To study the consistency of predictions across p-tools, here we present two indices and test them on five p-tools whose predictions are based on different types of background information. For this test, SNPs from well-known organism *Drosophila melanogaster* are considered.

Keywords: SNP · Gene functionality · Missense nonsense mutation

1 Introduction

A main factor underlying the conformation of proteins is their amino acid sequence. An individual nucleotide change, also called a Single Nucleotide Polymorphism (SNP), is a *missense mutation* when it causes a different protein, or a *nonsense mutation* when it causes a short and non-functional protein. The degree to which a SNP affects protein function is a key point, but its prediction remains an open problem.

Next-Generation Sequencing (NGS) technologies have made it possible to detect thousands of SNPs [1], but wet-lab studies needed to associate these SNPs with phenotypic traits are costly. To narrow down the list of candidate SNPs, several Bioinformatics tools, hereafter referred to as p-tools, have been developed to predict the impact of SNPs *in-silico*. P-tools can be based on information from amino acid sequences, protein structure, context, functional parameters and evolutionary information [2]. For instance, for sequence conservation

analysis, conserved amino acids—known to be relevant for protein function—are identified by alignment, and SNPs on these positions are identified as likely deleterious. Structure information is also used to infer sites with likely impact on protein function: SNPs in ligand-binding domains or active sites typically modify protein function. Based on this information, p-tools can be designed using either expert knowledge or machine learning techniques. P-tools not only vary in nature, but their outputs also vary in syntax and semantics, which makes comparison between them tricky. To tackle this problem, most strategies normalize predictions, forcing them into two classes, to evaluate classical performance metrics like accuracy, sensibility, sensitivity and ROC curves [4].

In this work, consistency across p-tools is evaluated by means of two proposed indices. For two given p-tools, the indices quantify the systematic disagreement between each pair of SNPs, i.e., count pairs of predictions ordered differently in each p-tool scale, without performing any scale normalization. An experimental study was carried out using five widely-used p-tools [3]. These were selected based on the diversity of knowledge or learning method they are based on, as well as the possibility to be run online with standard parameters. The consistency across p-tools was tested with SNPs from model organism *D. melanogaster*, a common starting point for data analysis.

2 Materials and Methods

The method to evaluate consistency across p-tools has two stages. The first one ponders, for each p-tool i , the order of two SNP effect predictions, (m_1, m_2) : m_1 can be more damaging than m_2 , the opposite can be true, or two mutations can be equally damaging. This preference relation is noted as follows:

- $m_1 \prec_i m_2$ if p-tool i considers m_1 to be less damaging than m_2 ;
- $m_1 \sim_i m_2 \iff \neg(m_1 \prec_i m_2) \wedge \neg(m_2 \prec_i m_1)$, if p-tool i cannot assert m_1 to be less or more damaging than m_2 .

To value the three possible orders for a pair m_1, m_2 , let $r_i(m_1, m_2)$ be defined as follows:

$$r_i(m_1, m_2) = \begin{cases} 1 & \text{if } m_1 \prec_i m_2 \\ -1 & \text{if } m_2 \prec_i m_1 \\ 0 & \text{if } m_2 \sim_i m_1 \end{cases} \quad (1)$$

The second stage values the degree of (dis)agreement of relative orders—given by all pairs of mutations—between two p-tools, i and j , through two indices (Eqs. 2 and 3): K_{all} is the ratio of mutations pairs ordered differently by both p-tools to all mutation pairs, considering all disagreements; K_{strong} is analogous but considers only opposite orderings.

$$K_{all} = \frac{|\{(m_1, m_2) \mid r_i(m_1, m_2) \neq r_j(m_1, m_2)\}|}{\binom{|M|}{2}} \quad (2)$$

$$K_{strong} = \frac{|\{(m_1, m_2) \mid r_i(m_1, m_2) \neq 0 \wedge r_i(m_1, m_2) = -r_j(m_1, m_2)\}|}{\binom{|M|}{2}} \quad (3)$$

For the case of discrete outputs, predictions are ordered according to labels, e.g. $\{\textit{benign}, \textit{possibly deleterious}, \textit{probably deleterious}\}$ and a preference relation: $\textit{benign} \prec \textit{possibly deleterious} \prec \textit{probably deleterious}$. When dealing with numerical outputs t_i , an inequality threshold δ_i is introduced, such that the preference relation for p-tool i is defined as follows: $m_1 \prec_i m_2 \iff t_i(m_2) - t_i(m_1) > \delta_i$, $\delta_i \geq 0$. Hence, $m_1 \sim_i m_2 \iff |t_i(m_2) - t_i(m_1)| \leq \delta_i$.

P-Tools: Selected p-tools have the following main features.

- PolyPhen2¹ uses protein sequences on a trained Naïve Bayes model to predict SNP sites which code for a protein’s structure or function.
- Provean² is based on a model that evaluates evolutionary information from protein sequence.
- Align-GVGD³ uses biophysical characteristics of amino acid and protein multiple sequence alignments on an evolutionary conservation model.
- Strum⁴ values changes in folding stability induced by SNPs based on a gradient boosting of Gibbs free-energy with different sequence and structure properties.
- Cupsat⁵ evaluates changes in protein stability induced by SNPs based on structure information of wild-type and mutant proteins.

Data: SNPs were analyzed on gene *vermillion*, locus *Dmel.CG2155*, on *D.mel*.

3 Results and Discussion

P-tools are compared pairwise with the K_{all} and K_{strong} indices and different equality thresholds. All possible SNPs in each sequence position of the *vermillion* gene are considered. See Fig. 1. Small index values represent similar pairwise outputs. For all pairwise comparisons, as the threshold increases, the K_{all} value first increases and then decreases when outputs became similar ($r_i(m_1, m_2) = 0$). **Polyphen2-Provean** follows that behavior after a threshold of 40% (data not shown). On the other hand, K_{strong} has a monotonically decreasing behavior. Two cases are possible when comparing pairwise p-tool outputs t_i , t_j with pairwise SNPs m_1 and m_2 : (1) $t_i(m_1) < t_i(m_2)$ and $t_j(m_1) < t_j(m_2)$ or (2) $t_i(m_2) < t_i(m_1)$ and $t_j(m_1) < t_j(m_2)$ ($t_i(m_1) \simeq t_i(m_2)$ or $t_j(m_1) \simeq t_j(m_2)$ are a middle step between these cases and are also analyzed). In case (1), there is an agreement according to K_{all} . When the threshold increases up to level $\delta_i = |t_i(m_1) - t_i(m_2)|$, $t_i(m_1) \simeq t_i(m_2)$ and $t_j(m_1) < t_j(m_2)$, making K_{all} count this as a disagreement. When the threshold reaches $\delta_j = |t_j(m_1) - t_j(m_2)|$, $t_j(m_1) \simeq t_j(m_2)$ and both tools agree again. In this case, the error will increase after δ_i and decrease after δ_j . Clearly, when δ is 100%, all pairs will be considered

¹ <http://genetics.bwh.harvard.edu/pph2/>.

² <http://sift.jcvi.org/>.

³ http://agvgd.hci.utah.edu/agvgd_input.php.

⁴ <https://zhanglab.ccmb.med.umich.edu/strum/>.

⁵ <http://cupsat.tu-bs.de>.

equal. In case (2), there is a disagreement according to K_{all} . After δ_i is reached, $t_i(m_1) \simeq t_i(m_2)$ and $t_j(m_1) < t_j(m_2)$, meaning they still disagree. Only after δ_j is reached, the two p-tools agree. In this case, the error decreases monotonically. An analogous analysis can be done with K_{strong} . In case (1), since both p-tools never give opposite results, K_{strong} evaluates outputs as an agreement, regardless of the threshold. In case (2), after δ_i is reached, the outputs are no longer opposite and are therefore equal according to K_{strong} . In both cases, the error can only decrease for increasing thresholds. While K_{all} has a monotonically decreasing behaviour with δ only in case (2), K_{strong} has such a behaviour for both cases, making it less sensitive to the equality threshold.

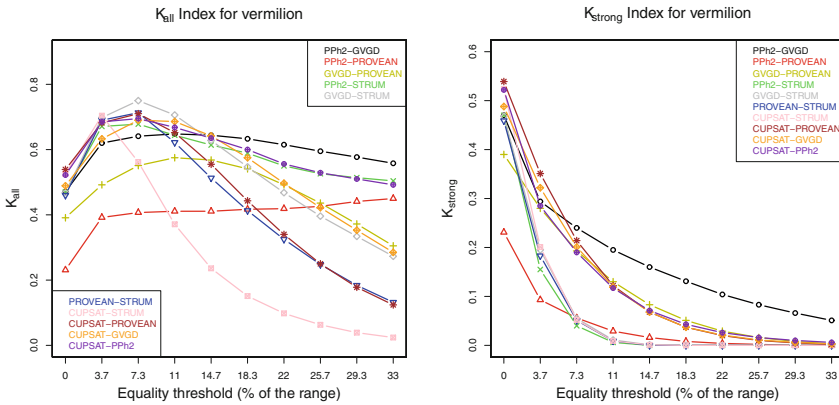


Fig. 1. Pairwise p-tool comparison. K_{all} (left) and K_{strong} (right), δ from 0% to 33%. Outputs scales: AlignGVGD [5, 215]; Provean [-15, 3]; PolyPhen-2 [0, 1]; Strum [-10, 11], and Cupsat [-23, 20], *D.mel.* gene *vermilion*

Note that even for $\delta \sim 10\%$, pairwise p-tool comparison with K_{strong} varies from 0.05% to 20% in the worst case. The two tools which agree the most across all δ are **Strum** and **Cupsat**, which makes sense since both work with similar knowledge, namely gene energy functions. On the other hand, the two tools which disagree the most are **PolyPhen2** and **Align-GVGD**, which also makes sense since one is based on structure and the other on evolutionary information.

4 Conclusions

Two indices were proposed to compare p-tools considering their most informative output. The indices do not require any normalization process. The comparison on *D.mel.* gene *vermilion* shows that predictions vary widely depending on the p-tool. Still, different outputs are not necessarily a problem, since they enable outputs to be integrated to achieve a more accurate prediction of SNP effects.

References

1. Wadapurkar, R., Vyas, R.: Computational analysis of next generation sequencing data and its applications in clinical oncology. *Inform. Med. Unlocked* **11**, 75–82 (2018)
2. Choi, Y., Sims, G.E., Murphy, S., Miller, J.R., Chan, A.P.: Predicting the functional effect of amino acid substitutions and indels. *PLOS ONE* **7**(10), e46688 (2012)
3. Thusberg, J., Olatubosun, A., Vihinen, M.: Performance of mutation pathogenicity prediction methods on missense variants. *Hum. Mutat.* **32**(4), 358–368 (2011)
4. Hicks, S., et al.: Prediction of missense mutation functionality depends on both the algorithm and sequence alignment employed. *Hum. Mutat.* **32**(6), 661–668 (2011)