Structural Connectivity Fingerprints – A new method of compound representation

***Krzysztof Rataj*[a], *Wojciech Czarnecki*[b], Sabina Podlewska[a,c], Andrzej J. Bojarski[a]**

[a]Institute of Pharmacology Polish Academy of Sciences, 12 Smętna Street, Kraków, Poland

[b]Faculty of Mathematics and Computer Science, Jagiellonian University, 6 Łojasiewicza Street, 30-348 Kraków, Poland

[c]Faculty of Chemistry Jagiellonian University 3 Ingardena Street 30-060 Krakow

e-mail: rataj@if-pan.krakow.pl

Introduction

Substructural key-based fingerprints, numerical representations of chemical compounds structure, are commonly used in drug design research. They are usually in a form of bitstring, where each bit depicts an occurrence of a specific substructure within the compound. These fingerprints allow for easy comparison of compound structures, and are used in classification tests. Having known structures of both active and inactive compounds for a given biological target, we can utilize fingerprints to predict the possibility of another compound to be a potential ligand for that protein. However, the key-based fingerprints have also their flaws, as multiple compounds can share identical or very similar fingerprint, which in turn may lead to misclassification errors.

To address this issue, we designed a new substructural fingerprint – the Substructural Connectivity Fingerprint (SCFP).

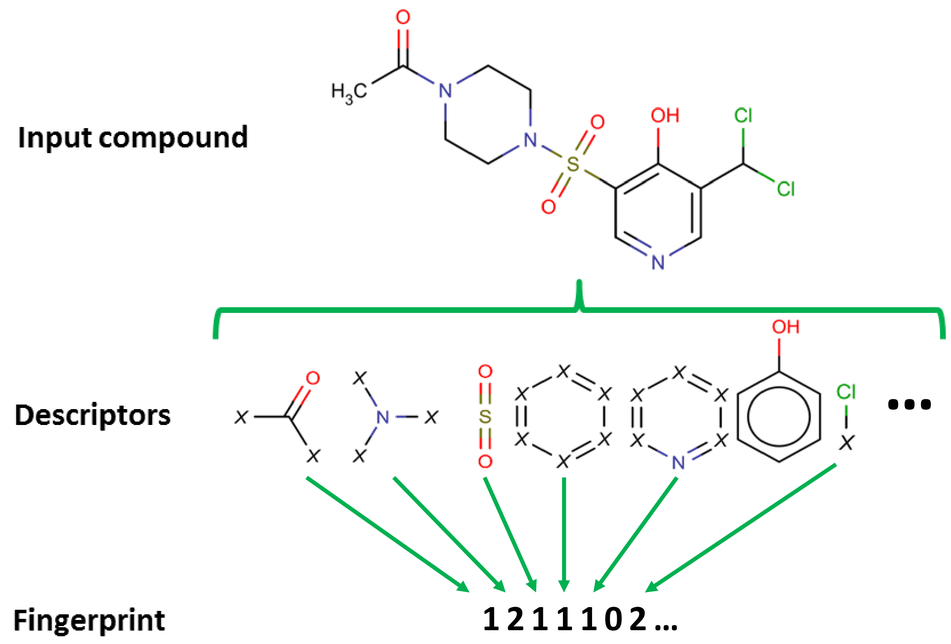

Figure 1.: Visualisation of fingerprint generation

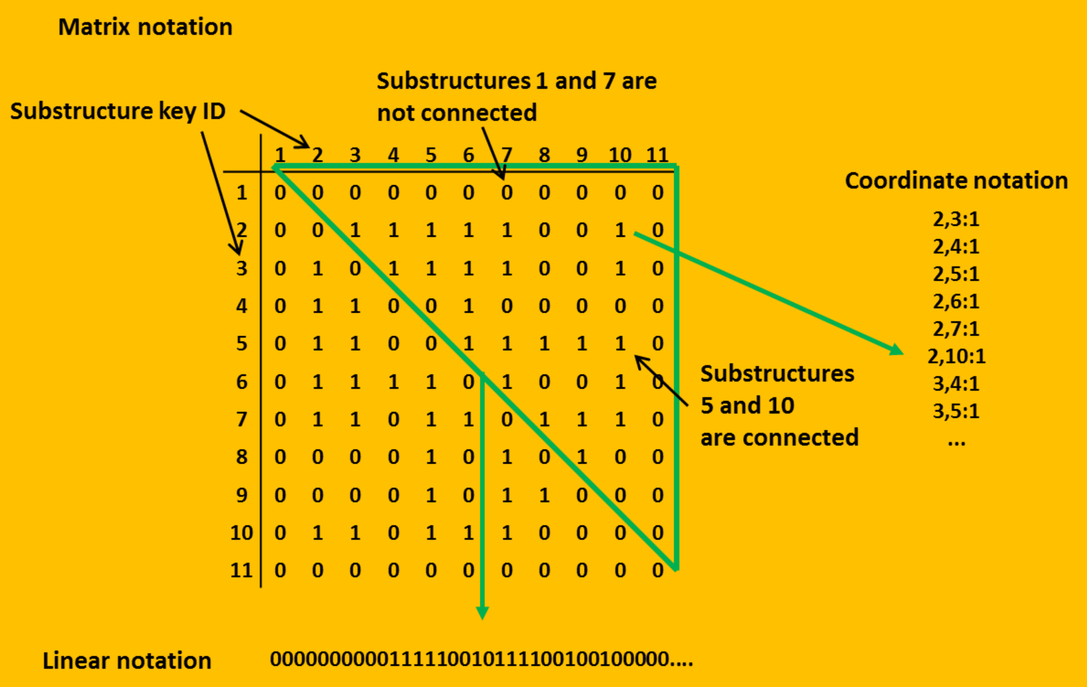

Figure 2.: Possible notations of the SCFP.

What is SCFP?

The SCFP is an upgrade to standard key-based substructural fingerprints, which takes into consideration the connections between particular substructures within the compound. The data is stored in a form of a 2D matrix or coordinates of non-zero bits. The substructure keys used for its creation can be supplied from any of the existing key-based fingerprints (Klekota-Roth FP[1], MACCS[2], Substructure FP) or manually using SMARTS expressions.

How to analyze it?

Regular key-based substructural fingerprints can be analyzed in various ways, including Machine Learning (ML), clustering or similarity searching algorithms. Since the SCFP can also be transformed into a bit string, it can be easily processed with the same methods as regular fingerprints, even though it may take more time to do so (linear SCFP contains $N^2/2$ bits compared to regular fingerprints). However, after proper implementation, the coordinate version of the SCFP is processed in comparable times to its regular counterparts.

Time for tests

To test the capabilities of the newly developed fingerprint, a series of 5-fold cross-validation classification tests were performed. To do so, we extracted known active and inactive compounds for 9 GPCR and 5 protein kinase targets from the ChEMBL database. For each of those sets, regular key-based and SCFP fingerprints were calculated, using 3 substructure key sets: MACCS, Substructure FP, Klekota-Roth FP.

The fingerprints were analyzed using various available ML methods, such as Support Vector Machines (SVM), Naive Bayes (NB) and Random Forest (RF). Additionally, a novel methodology, Extreme Entropy Machines (EEM[3]), was implemented to help analyze the SCFPs. All methods were optimized towards Balanced Accuracy metric (BAC).

The results clearly show, that adding the connectivity data in SCFP greatly increased the BAC values of the classification tests, which is especially visible in the case of Klekota-Roth-based SCFP. What is more, the addition of EEM methodology further increased the accuracy of the classification.

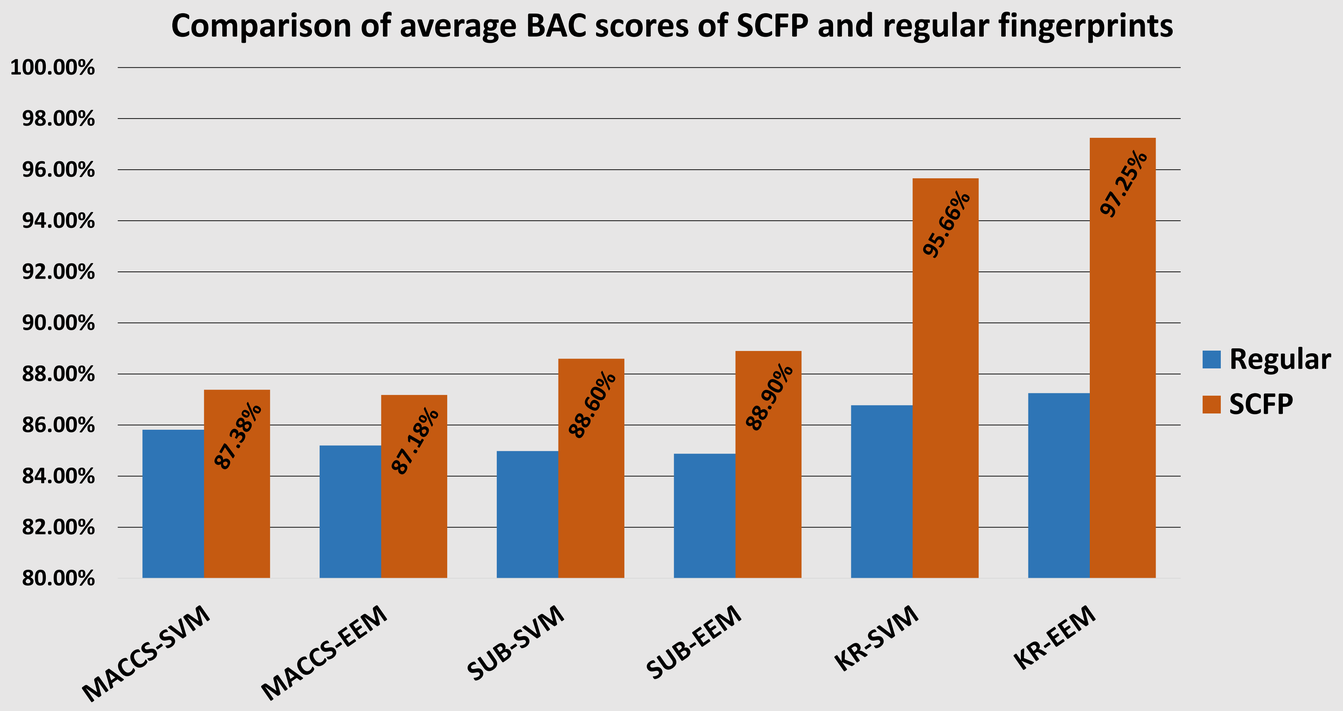

Figure 3.: Average Balanced Accuracy scores for compound classification tests. Only two ML methods are shown: Support Vector Machines (SVM) and Extreme Entropy Machines (EEM). Three substructure key sets were used: MACCS, Substructure FP (SUB) and Klekota-Roth FP (KR).

References:

1. Klekota, J. & Roth, F. P. Bioinformatics 24, 2518–25 (2008)
2. Ewing, T., Baber, J. C. & Feher, M. , J. Chem. Inf. Model. 46, 2423–2431
3. Czarnecki WM, Tabor J, 2015, Pattern Anal. Appl.