Supramolecular Decoration of Metal-Organic Hosts From Products of Template-Directed Synthesis

<u>Tamara D. Hamilton</u>, Tomislav Friscic, Leonard R. MacGillivray, *Department of Chemistry, University of Iowa, Iowa City, IA, USA*. E-mail: tamara-hamilton-1@uiowa.edu

Metal-mediated self-assembly is the process whereby organic subunits come together in solution by way of coordination bonding to metal centers to form well-defined infinite (i.e. coordination polymers, metal-organic frameworks) or discrete (i.e. metal-organic polygons, polyhedra) aggregates. Metal-organic polygons and polyhedra have found utility in the last year in the fields of host-guest chemistry and materials science. Congruous design of the organic subunit is imperative in metal-mediated self-assembly, as the subunit must contain appropriate functionalities to recognize the metal in such a way that the desired aggregate will form. Most self-assembling systems studied to date have been based on symmetrical ligands, those with multiple copies of identical binding functionalities. We have used template-directed synthesis in the solid state to produce unsymmetrical tetrapyridyl cyclobutane molecules in quantitative yield and in gram quantities. When these polyfunctional molecules self-assemble with metals, they give rise to metal-organic polyhedra [1]. In addition, the outer surfaces of these polyhedra can be decorated in a supramolecular fashion with various organic groups, opening doors to applications in anion transport and biomedicinal chemistry. That template-directed synthesis is used to produce molecules otherwise difficult or impossible to obtain, and that these molecules, in turn, assemble with metals to produce interesting and useful metalorganic assemblies augurs well for the future study of the crystalline solid state as a reaction medium.

[1] Hamilton T. D., Papaefstathiou G. S., MacGillivray L. R., J. Am. Chem. Soc., 2002, 124, 11606.

Keywords: metal-organic complexes, solid-state photochemistry, self-assembly supramolecular chemistry