# Targeting AKAP12/Gravin Using Cloud-Computing for Drug Discovery and Molecular Simulation

**Sashank Shorey, Dr. Bradley McConnell**

[1]University of Houston College of Pharmacy; Houston, TX

Contact Information:
Sashank Shorey
University of Houston
Phone: (972)7575887
Email: ssshorey@uh.edu

## ABSTRACT

**Background:** AKAP12/Gravin signalosome is a critical protein involved in regulating cardiac function by regulating intracellular signaling through its binding, in part, to $\beta_2$-adrenergic receptors ($\beta_2$ARs), protein kinase A (PKA), and protein kinase (PKC). In this study, we characterize AKAP12/Gravin protein binding inhibition by using Cloud-Computing for the purpose of performing docking simulations for drug discovery.

**Methods:** This project involved creating Python/PySpark compatible scripts for use with PySpark and Dataframes on Linux Virtual Machines, as well as optimizing parallelization and specifications of the Virtual Machines in use for the various drug discovery and molecular simulation programs.

## BACKGROUND

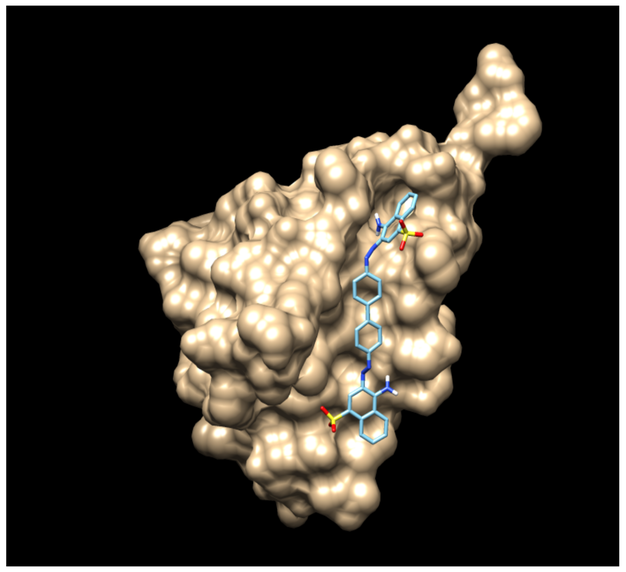
Emerging technology's newest evolution, "Cloud-Computing", provides opportunities to study pharmacological leads and mechanisms. Cloud-Computing can be viewed as an abstraction to traditional computer clusters. It implements a conceptual layer that divides the hardware from the software. Cloud providers like Microsoft-Azure, Amazon-AWS, and Google-Cloud-Platform allow Virtual Machines running Linux to be configured as abstract clusters tailored for our applications and sharing resilient data with a RESTful (Representational-State-Transfer) architecture that integrates our high-performance computational pipeline with web services. This project optimizes our drug discovery pipeline, utilizing several programs; including AutoDock, Chimera, Firefly, and GROMACS, in the "Cloud"

## METHODS

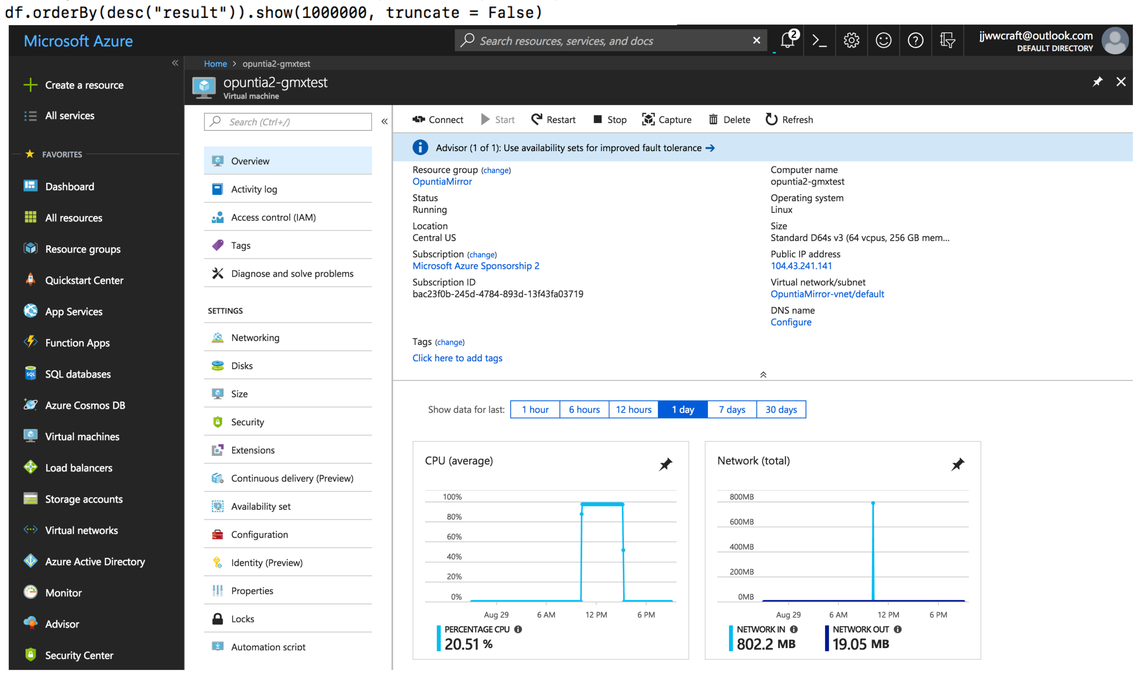
**Study Design**

- We chose to use D-series Virtual Machines. These Virtual Machines offer good CPU-to-memory balance and are easy to scale up for larger workloads. Standard D64s v3 (64 vcpus, 256 GB memory) virtual machines running Linux
- Using Azure File Storage as the standard for data transfer/storage offers simplified integration and shared access for any number of different Virtual Machines.
- The Azure File Storage component presents the largest bottleneck for our VM performance
- Measured the efficiency in time, performance and accuracy of using high-performance virtual machines (with nodes run in parallel) through GROMACS molecular simulations

```
from pyspark import SparkConf, SparkContext
from pyspark.sql import SQLContext
from pyspark.sql.functions import desc
import sys
import os

sc = SparkContext()
sqlContext = SQLContext(sc)

arecord = []

os.chdir("MAY3/strategy1")
for record in open("LS-full"):
 if "pdbqt" in record:
   cell = record.split()

   with open(cell[8]) as file:
     for line in file:
       if "RESULT" in line:
         result = line.split()
         newrecord = (cell[4], cell[8], result[3])
         arecord.append([cell[4], cell[8], result[3]])

         file.close()
         break

df = sqlContext.createDataFrame(arecord , ["number", "file", "result"])
df.orderBy(desc("result")).show(1000000, truncate = False)
```



| number | file | result |
|---|---|---|
| 24498 | MayHitDiscoverEE10016-dock.pdbqt | -8.1 |
| 31572 | MayHitDiscoverBB10015-dock.pdbqt | -7.9 |
| 23904 | MayHitDiscoverCC17211-dock.pdbqt | -7.9 |
| 31572 | MayHitDiscoverBB10014-dock.pdbqt | -7.8 |
| 24624 | MayHitDiscoverCC10904-dock.pdbqt | -7.8 |
| 27675 | MayHitDiscoverBB10005-dock.pdbqt | -7.7 |
| 31572 | MayHitDiscoverBB10013-dock.pdbqt | -7.7 |
| 24795 | MayHitDiscoverCC7432-dock.pdbqt | -7.7 |
| 24795 | MayHitDiscoverCC7433-dock.pdbqt | -7.7 |
| 23607 | MayHitDiscoverBB7842-dock.pdbqt | -7.6 |
| 27675 | MayHitDiscoverBB10006-dock.pdbqt | -7.6 |
| 24075 | MayHitDiscoverCC3832-dock.pdbqt | -7.6 |
| 26829 | MayHitDiscoverCC10013-dock.pdbqt | -7.6 |
| 24795 | MayHitDiscoverDD2788-dock.pdbqt | -7.6 |
| 26829 | MayHitDiscoverCC10012-dock.pdbqt | -7.6 |
| 23904 | MayHitDiscoverEE4730-dock.pdbqt | -7.6 |
| 23481 | MayHitDiscoverCC3838-dock.pdbqt | -7.5 |
| 31572 | MayHitDiscoverBB10012-dock.pdbqt | -7.5 |
| 24624 | MayHitDiscoverCC10905-dock.pdbqt | -7.5 |
| 26829 | MayHitDiscoverCC10014-dock.pdbqt | -7.5 |




- The PySpark implementation of Apache Spark's large scale data parallelization framework offers much faster workload processing for drug discovery
- AutoDock Vina, ADT, and Chimera (used to model the protein/ligand complex as shown on left), processed large amounts of data.
- PySpark's integration of DataFrames allows for highly parallelized data with defined structures (the DataFrame shown in the table to the left lists the pseudo free energy binding values for ligand-protein docking).
- Microsoft Azure's highly scalable and powerful Virtual Machines allow for efficient use of many worker nodes.
- Azure offers an HDInsight Virtual Machine, with built-in Apache Spark/Hadoop/PySpark support for integration of big data parallelization with Virtual Machines
- In order to fully benefit from big data parallelization and use of Virtual Machines, computational scripts using the Drug Discovery programs (ie. GROMACS and AutoDockTools) must be written in PySpark and have DataFrame capability. An example of a free energy binding script is shown, along with its DataFrame result.

## CONCLUSIONS

- Microsoft Azure's Cloud computing service and PySpark's parallelization framework offer simplified data integration, and scalable architecture for large computational jobs
- However, the performance of Azure File Storage and data access presents a major obstacle
- Future directions include troubleshooting the Azure File Share bottleneck, and integrating PySpark parallelization into existing Drug Discovery Pipeline shell/Python computational scripts and tools.