CONTENTS:

1)HOW TO RUN THE CODE:

1) The mat file needs to be opened as the variables will be used later.
2) From the GSP toolbox, execute:
    gsp_start;
    gsp_make;
   In order to use commands from the gspbox.
3) Run the matlab command:
    addpath 'directory'
   to add the 'Sampling', 'Local sets'\ , 'Reconstruction Algos'\ directories
4) The codes errorfix.m and placeholder.m are used to generate the plots of error versus iterations
   You can run these codes first to get a general idea on what to do
   (Instructions for inputs/outputs of each code are given in the CODES EXPLAINED section)
5) Each folder contains the supporting codes (Reconstruction)

2)VARIABLES EXPLAINED:

Open the .mat file provided in the repo
   rang: 50 node random regular graph, each node connected to 3 others
   rangi: 10 node random regualr graph, each node connected to 2 others
   localset_bfs: localset generated using BFS algorithm
   localset_onehop: localset generated using One Hop algorithm
   localset_graphall: localset generated using Johnsons (graphallshortestpath) algorithm
   G_minne: Minnesota road graph.
   x: 50 node graph signal with sharp cutoff frequency of 3.
   Smin: 30 node sampling set for rang with cutoff frequency 3
   Sonehop: onehop sampling set for rang

3)CODES EXPLAINED:

Each folder contains its corresponding codes. Please run addpath before running any code

LOCAL SETS:

Each local set is represented as a matrix.
If localset[i][j]=1, then j belongs to i's localset.
The sum along each column of the matrix returns 1 (since local sets are mutually exclusive)
The sum along each row returns the number of nodes in that node's localset.

bfslocalset.m:
   Generates a local set based on our BFS algorithm

inputs: graph G, sampling set S
outputs: localset

graphallshortestpath.m:
Generates a local set based on Johnson's algorithm.
inputs: graph G, sampling set S
outputs: localset

SAMPLING:
Onehop Sampling set: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7055883
All the algorithsm can be found in the paper: https://arxiv.org/pdf/1507.08822.pdf

maxfrobnorm.m:
Generates a sampling set that Maximizes the Frobenius Norm
inputs: graph G, cutoff frequency w, number of samples M
outputs: sampling set S

maxsigmin.m:
Generates a sampling set that maximizes the minimum singular value
inputs: graph G, cutoff frequency w, number of samples M
outputs: sampling set S

maxvolume.m:
Generates a sampling set that Maximizes the volume of parallelopiped
inputs: graph G, cutoff frequency w, number of samples M
outputs: sampling set S

minfrobnorm.m:
Generates a sampling set that Minimizes the Frobenius Norm (MinPinv)
inputs: graph G, cutoff frequency w, number of samples M
outputs: sampling set S

minuniset.m:
Generates the smallest sampling set that allows unique recovery
inputs: graph G, cutoff frequency w, number of samples M
outputs: sampling set S

randsamp.m:
Generates a random sampling set
inputs: graph G, cutoff frequency w, number of samples M
outputs: sampling set S

onehop.m:
Generates the onehop sampling set and corresponding localsets

inputs: graph G
outputs: sampling set S, localset

RECONSTRUCTION ALGOS:

Input the unsampled signal for each of these codes, sampling takes place within the code.

ilsr.m:

Iterative least squares reconstruction
inputs: graph G, sampling set S, unsampled signal f, cutoff freq w, # of iterations iter
outputs: reconstructed signal f1

iwr.m:

Iterative Weighting Resonctruction
inputs: graph G, sampling set S, localset, unsampled signal f, cutoff freq w, # of iterations iter
outputs: reconstructed signal f1

ipr.m:

Iterative Propagating Resonctruction
inputs: graph G, sampling set S, localset, unsampled signal f, cutoff freq w, # of iterations iter
outputs: reconstructed signal f1

MISC:

pwproject.m:

Paley Weiner Projection
inputs: graph G, signal f, cutoff frequency w
outputs: signal f

makesignal.m:

inputs: graph G, cutoff frequency w, a constant const
outputs: a randomly generated signal x

placeholder.m:

Generates reconstruction errors for ilsr, ipr, iwr
Averaged over 'iter' graphs and 'j' signals
Keep the number of samples>(total # of nodes)/2
inputs: cutoff frequency w, number of samples M
outputs: none, plots error vs number of iterations

errorfix.m:

Does the same thing as placeholder.m, except for a single signal and single graph.
This allows you to manipulate what the graph and signal will be
inputs: graph G, signal x, cutoff freq w
outputs: none, error vs number of iterations