In [12]:

```
from utility import *

from collections import Counter
from scipy.stats import ks_2samp

import matplotlib.pyplot as plt
import numpy as np
import random
import pandas as pd
import seaborn as sns

%matplotlib inline
```

Dataset Loading

The data sets needed for the loaders can be found at snap.stanford.edu/decagon. The side effect information was curated from the TWOSIDES, OFFSIDES, and Sider databases.

In [13]:

```
combo2stitch, combo2se, se2name = load_combo_se()
net, node2idx = load_ppi()
stitch2se, se2name_mono = load_mono_se()
stitch2proteins = load_targets(fname='bio-decagon-targets-all.csv')
se2class, se2name_class = load_categories()
se2name.update(se2name_mono)
se2name.update(se2name_class)
```

```
Reading: bio-decagon-combo.csv
Drug combinations: 63473 Side effects: 1318
Drug-drug interactions: 4651131
Reading: bio-decagon-ppi.csv
Edges: 715612
Nodes: 19081
Reading: bio-decagon-mono.csv
Reading: bio-decagon-targets-all.csv
Reading: bio-decagon-effectcategories.csv
```

Basic Statistics

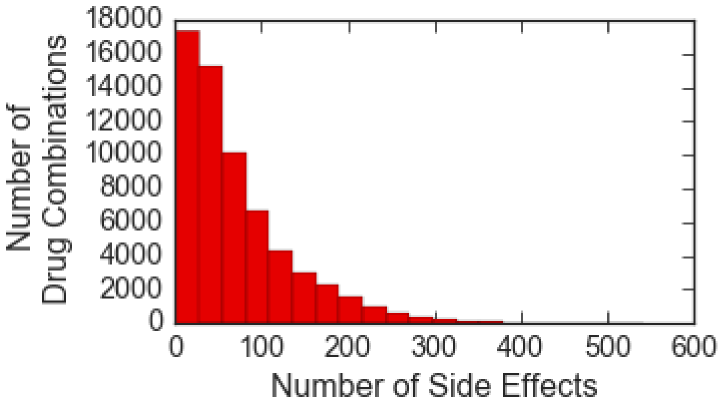
How many side effects does each drug combination have?

In [14]:

```
def plot_distribution(dist, title="", x_label="", y_label="", file_name=Non
    plt.figure(figsize=(6, 3.5))
    sns.set_context("paper", font_scale=1.8)
    sns.set_style('ticks')
    sns.set_style({"xtick.direction": "in", "ytick.direction": "in"})
    sns.distplot(dist, kde=False, color=sns.xkcd_rgb['red'], bins=20, hist_
    plt.xlabel(x_label)
    plt.title(title)
    plt.tight_layout()
    plt.gcf().subplots_adjust(left=0.2, right=0.8, top=0.8, bottom=0.2)
    plt.ylabel(y_label)
    if file_name:
        plt.savefig(file_name)
```

```
In [15]: distribution_combos = [len(combo2se[combo]) for combo in combo2se]
         print "Median number of side effects per drug combination", np.median(distr
         plot_distribution(distribution_combos, "", "Number of Side Effects", "Numbe
```

```
Median number of side effects per drug combination 53.0
```



How frequent are different side effects?

In [16]:

```
from IPython.display import display, HTML

def get_se_counter(se_map):
    side_effects = []
    for drug in se_map:
        side_effects += list(set(se_map[drug]))
    return Counter(side_effects)

combo_counter = get_se_counter(combo2se)

print("Most common side effects in drug combinations:")
common_se = []
common_se_counts = []
common_se_names = []
for se, count in combo_counter.most_common(20):
    common_se += [se]
    common_se_counts += [count]
    common_se_names += [se2name[se]]
df = pd.DataFrame(data={"Side Effect": common_se, "Frequency in Drug Combos
display(df)
```

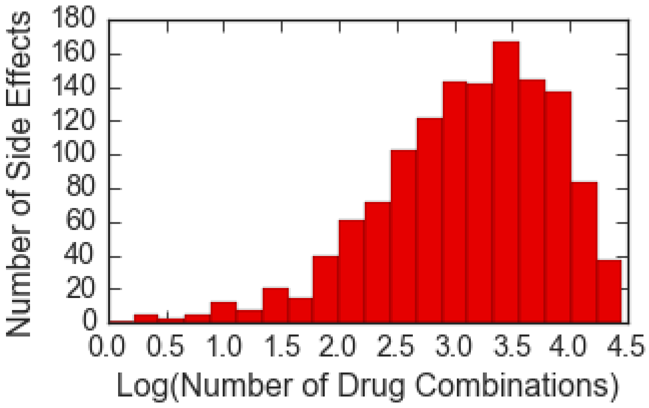
```
Most common side effects in drug combinations:
```

Out[ ]:

	Frequency in Drug Combos	Name	Side Effect
0	28568	arterial pressure NOS decreased	C0020649
1	27006	anaemia	C0002871
2	26037	Difficulty breathing	C0013404
3	25190	nausea	C0027497
4	24430	neumonia	C0032285
5	24260	Fatigue	C0015672
6	23894	Pain	C0030193
7	23848	diarrhea	C0011991
8	23515	asthenia	C0004093
9	23043	emesis	C0042963
10	21981	edema extremities	C0085649
11	21806	body temperature increased	C0015967
12	21781	pleural pain	C0008033
13	21410	abdominal pain	C0000737
14	21322	Hypoventilation	C0398353
15	21013	chest pain	C0008031
16	20204	dizziness	C0012833
17	19930	Back Ache	C0004604
18	19803	Head ache	C0018681
19	19376	High blood pressure	C0020538

Plot of Side Effect Frequency

In [17]:
```
plot_distribution(np.log10(np.asarray(list(zip(*combo_counter.items())[1]))
```



Side Effect Cooccurrence in Drug Combinations

In [18]:
```
combos = combo2se.keys()
combo_probability_distribution = np.asarray([len(combo2se[combo])*1.0 for c
combo_probability_distribution = combo_probability_distribution/np.sum(comb

se2combo = defaultdict(set)
for combo in combo2se:
    for se in combo2se[combo]:
        se2combo[se].add(combo)
```

We observe that polypharmacy side effects do not appear independently of one another in co-prescribed drug pairs (\ie, drug combinations), suggesting that joint modeling over multiple side effects can aid in the prediction task. To quantify the co-occurrence between side effects, we count the number of drug combinations in which a given side effect co-occurs with other side effects, and then use permutation testing with a null model of random co-occurrence. As exemplified for hypertension and nausea below, we find that the majority of the most common side effects are either significantly overrepresented or underrepresented with respect to how often they co-occur with nausea/hypertension as side effects in drug combinations, at $\alpha = 0.05$.

In [19]:
```
# Permutation test testing the significancy between the drug combinations a
# as compared to other common side effects
def run_permutation_test(se_oi, num_permutations = 2000):
    se_oi_combos = se2combo[se_oi]
    side_effects = []
    names = []
    real_overlaps = []
    mean_permuted_overlap = []
    probabilities = []
    for se, count in combo_counter.most_common(51):
        if se == se_oi:
            continue
        real_combos = se2combo[se]
        real_overlap = len(real_combos.intersection(se_oi_combos))
        permuted_overlaps = []
        for i in range(num_permutations):
            combo_sample = np.random.choice(combos, len(real_combos), repla
            permuted_overlaps += [len(se_oi_combos.intersection(set(combo_s
        probability = np.sum(np.asarray(permuted_overlaps) >= real_overlap)
        side_effects += [se]
        names += [se2name[se]]
        real_overlaps += [real_overlap]
        mean_permuted_overlap += [np.mean(permuted_overlaps)]
        probabilities += [probability]
    df = pd.DataFrame(data={"Side Effect": side_effects, "True Overlap": re
    df = df[['Side Effect', 'Name', 'True Overlap', 'Mean Permuted Overlap'
    display(df)
```

In [20]:

```
# For hypertension
run_permutation_test('C0020538')
```

Out[ ]:

	Side Effect	Name	True Overlap	Mean Permuted Overlap	Probability True < Permuted
0	C0020649	arterial pressure NOS decreased	10557	11168.6310	1.0000
1	C0002871	anaemia	9457	10625.5935	1.0000
2	C0013404	Difficulty breathing	9974	10284.1575	1.0000
3	C0027497	nausea	9326	9983.4730	1.0000
4	C0032285	neumonia	9032	9710.2255	1.0000
5	C0015672	Fatigue	9169	9648.1485	1.0000
6	C0030193	Pain	9804	9517.4920	0.0000
7	C0011991	diarrhea	8625	9499.6700	1.0000
8	C0004093	asthenia	9150	9379.2765	1.0000
9	C0042963	emesis	8227	9205.6620	1.0000
10	C0085649	edema extremities	9011	8818.5175	0.0000
11	C0015967	body temperature increased	7536	8752.6295	1.0000
12	C0008033	pleural pain	8941	8741.9640	0.0000
13	C0000737	abdominal pain	8923	8605.4555	0.0000
14	C0398353	Hypoventilation	9926	8571.8495	0.0000
15	C0008031	chest pain	9986	8458.9515	0.0000
16	C0012833	dizziness	8409	8155.1085	0.0000
17	C0004604	Back Ache	8879	8053.4790	0.0000
18	C0018681	Head ache	8402	8006.1115	0.0000
19	C0009676	confusion	7311	7840.3240	1.0000
20	C0011175	dehydration	6877	7740.0310	1.0000
21	C0003467	Anxiety	8695	7695.4900	0.0000
22	C0035078	kidney failure	7261	7660.8715	1.0000
23	C0043096	loss of weight	7523	7621.0155	0.9735
24	C0020456	hyperglycaemia	7886	7617.4375	0.0000
25	C0013604	edema	7163	7482.2830	1.0000
26	C0003123	Anorexia	6811	7388.9615	1.0000
27	C0003862	Aching joints	7994	7344.9990	0.0000
28	C0022660	acute kidney failure	6381	7302.7290	1.0000
29	C0442874	neuropathy	7938	7286.8275	0.0000
30	C0030196	Extremity pain	7676	7247.0995	0.0000
31	C0042029	Infection Urinary Tract	7502	7192.1085	0.0000
32	C0010200	Cough	7260	7186.2450	0.0625
33	C0009806	constipated	7481	7183.9235	0.0000
34	C0039231	heart rate increased	7089	7156.0390	0.9260
35	C0043094	weight gain	7621	6981.6530	0.0000
36	C0040034	thrombocytopenia	5300	6961.4105	1.0000
37	C0032227	Pleural Effusion	6137	6900.5185	1.0000
38	C0243026	sepsis	5680	6773.1200	1.0000
39	C0018802	Cardiac decompensation	7612	6766.7475	0.0000
40	C0027051	heart attack	8271	6692.5900	0.0000
41	C1145670	respiratory failure	5855	6536.9340	1.0000

	Side Effect	Name	True Overlap	Mean Permuted Overlap	Probability True < Permuted
42	C0004238	AFIB	6278	6386.6380	0.9880
43	C0013144	drowsiness	6195	6374.8605	1.0000
44	C0015230	eruption	5653	6367.9340	1.0000
45	C0917801	insomnia	6756	6339.0310	0.0000
46	C0037763	muscle spasm	6867	6290.1030	0.0000
47	C0038454	apoplexy	8084	6253.6085	0.0000
48	C0027947	Neutropenia	4426	6228.8030	1.0000
49	C0020625	blood sodium decreased	5353	6031.0370	1.0000

In [21]:

```
# For nausea
run_permutation_test('C0027497')
```

Out[ ]:

	Side Effect	Name	True Overlap	Mean Permuted Overlap	Probability True < Permuted
0	C0020649	arterial pressure NOS decreased	13623	13817.6665	1.0000
1	C0002871	anaemia	12668	13141.3570	1.0000
2	C0013404	Difficulty breathing	12823	12715.3290	0.0215
3	C0032285	neumonia	11075	11998.8955	1.0000
4	C0015672	Fatigue	13570	11921.7745	0.0000
5	C0030193	Pain	12699	11760.4490	0.0000
6	C0011991	diarrhea	13492	11740.9805	0.0000
7	C0004093	asthenia	13137	11587.6845	0.0000
8	C0042963	emesis	16363	11377.2805	0.0000
9	C0085649	edema extremities	11139	10890.9000	0.0000
10	C0015967	body temperature increased	10861	10811.9245	0.1760
11	C0008033	pleural pain	11890	10799.2440	0.0000
12	C0000737	abdominal pain	12145	10627.1125	0.0000
13	C0398353	Hypoventilation	11109	10590.1065	0.0000
14	C0008031	chest pain	11003	10448.3195	0.0000
15	C0012833	dizziness	11644	10073.0220	0.0000
16	C0004604	Back Ache	11152	9944.3010	0.0000
17	C0018681	Head ache	11346	9883.8600	0.0000
18	C0020538	High blood pressure	9326	9689.2270	1.0000
19	C0009676	confusion	9862	9681.1065	0.0005
20	C0011175	dehydration	10291	9557.3490	0.0000
21	C0003467	Anxiety	10263	9503.0730	0.0000
22	C0035078	kidney failure	8367	9461.2210	1.0000
23	C0043096	loss of weight	10683	9411.3420	0.0000
24	C0020456	hyperglycaemia	9321	9408.8050	0.9595
25	C0013604	edema	8827	9240.2495	1.0000
26	C0003123	Anorexia	11131	9124.1175	0.0000
27	C0003862	Aching joints	9981	9071.0055	0.0000
28	C0022660	acute kidney failure	8075	9016.2585	1.0000
29	C0442874	neuropathy	9087	9001.2190	0.0470
30	C0030196	Extremity pain	9742	8948.4745	0.0000
31	C0042029	Infection Urinary Tract	9281	8879.6780	0.0000
32	C0010200	Cough	9406	8874.1120	0.0000
33	C0009806	constipated	10229	8871.6220	0.0000
34	C0039231	heart rate increased	8770	8837.0945	0.9020
35	C0043094	weight gain	8956	8620.5610	0.0000
36	C0040034	thrombocytopenia	7512	8596.8345	1.0000
37	C0032227	Pleural Effusion	7840	8516.5250	1.0000
38	C0243026	sepsis	7473	8367.0850	1.0000
39	C0018802	Cardiac decompensation	7753	8354.2405	1.0000
40	C0027051	heart attack	7721	8267.0380	1.0000
41	C1145670	respiratory failure	7169	8072.5870	1.0000

	Side Effect	Name	True Overlap	Mean Permuted Overlap	Probability True < Permuted
42	C0004238	AFIB	6995	7886.3295	1.0000
43	C0013144	drowsiness	7976	7871.2930	0.0160
44	C0015230	eruption	7785	7860.9235	0.9425
45	C0917801	insomnia	8845	7825.6305	0.0000
46	C0037763	muscle spasm	8537	7769.9140	0.0000
47	C0038454	apoplexy	7332	7722.1020	1.0000
48	C0027947	Neutropenia	6996	7691.7345	1.0000
49	C0020625	blood sodium decreased	7267	7447.0185	1.0000

How similar are the drug target profiles of drug combinations?

Third, we probe the relationship between proteins targeted by a drug pair and occurrence of side effects. Let T_i represent a set of target proteins associated with drug i, we then calculate the Jaccard similarity between target proteins of a given drug pair (i, j) as:
$\text{Jaccard}(i, j) = |T_i \cap T_j|/|T_i \cup T_j|$.
We see most drug combinations have zero target proteins in common, random drug pairs have smaller overlap in targeted proteins than co-prescribed drugs, andthat this trend is unequally observed across different side effects.

In [23]:

```
def jaccard(set1, set2):
    num = len(set(set1).intersection(set(set2)))
    den = len(set(set1).union(set(set2)))
    return num*1.0/den

# Only examining those drugs we have drug target information for
valid = []
for stitch in stitch2se:
    if len(stitch2proteins[stitch]) > 0:
        valid += [stitch]

# Jaccard similarity between drug target profiles of drugs in drug combinat
jaccard_combos = {}
for combo in combo2se:
    stitch1, stitch2 = combo2stitch[combo]
    if stitch1 in valid and stitch2 in valid:
        jaccard_combos[combo] = jaccard(stitch2proteins[stitch1], stitch2pr

# Jaccard similarity between drug target profiles of random drugs
jaccard_random = []
for i in range(len(jaccard_random)):
    stitch1 = np.random.choice(valid, 1, replace=False)[0]
    stitch2 =  np.random.choice(valid, 1, replace=False)[0]
    jaccard_random += [jaccard(stitch2proteins[stitch1], stitch2proteins[st
jaccard_random = np.asarray(jaccard_random)
```

In [24]:

```
import pandas as pd

def plot_jaccard_distribution_multiple(ses):
    group_names = {'Random drug pairs': jaccard_random, 'All drug combinati
    order = ['Random drug pairs', 'All drug combinations'] + [nicknames[se]
    for se in ses:
        se_combos = se2combo[se].intersection(set(jaccard_combos.keys()))
        in_jaccard = np.asarray([jaccard_combos[combo] for combo in se_comb
        group_name = nicknames[se]
        group_names[group_name] = in_jaccard
    categories = {'No shared drug targets': (-.01, 0), 'Fewer than 50% shar
    groups, similarities, fractions = [], [], []
    for name in group_names:
        arr = group_names[name]
        for category in categories:
            min_val, max_val = categories[category]
            value = np.sum((arr > min_val) * (arr <= max_val))*1.0/len(arr)
            groups += [name]
            similarities += [category]
            fractions += [value]
    data = pd.DataFrame({ '' : groups, 'Jaccard Similarity Between Drug Tar
    plt.figure(figsize=(3, 5))
    sns.set_context("paper", font_scale=6)
    sns.set_style('ticks')
    sns.set_style({"xtick.direction": "in", "ytick.direction": "in"})
    g = sns.factorplot(x="Jaccard Similarity Between Drug Target Profiles",
                  size=18, kind="bar", palette=['#535456', '#9ea3a8', '#34
    plt.tight_layout()
    plt.xlabel('')
    plt.savefig('multiple_dist.pdf')
```

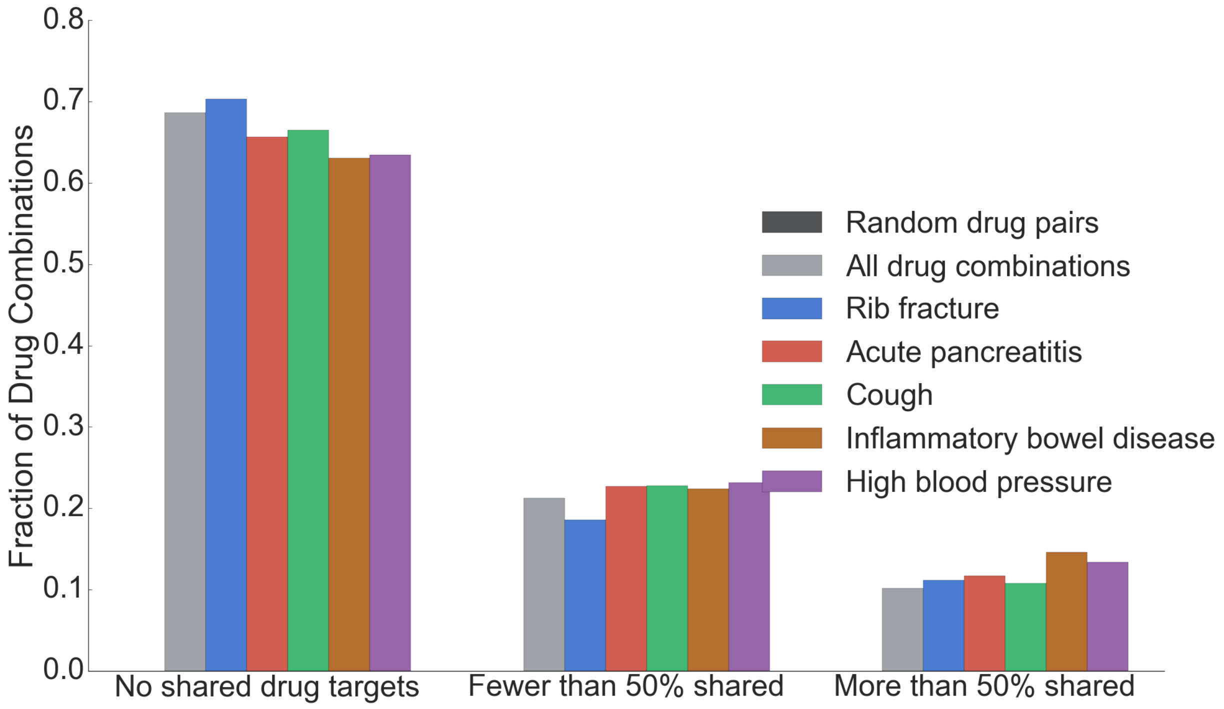
In [25]:

```
nicknames = {'C0035522': 'Rib fracture', 'C0001339': 'Acute pancreatitis',

plot_jaccard_distribution_multiple(['C0035522', 'C0001339',  'C0010200', 'C
```

```
/Users/monicaagrawal/anaconda/lib/python2.7/site-packages/ipykernel/__mai
n__.py:17: RuntimeWarning: invalid value encountered in double_scalars
/Users/monicaagrawal/anaconda/lib/python2.7/site-packages/seaborn/categor
ical.py:3304: UserWarning: The `x_order` parameter has been renamed `orde
r`
  UserWarning)

<matplotlib.figure.Figure at 0x1a33346e50>
```



In [ ]: