Exploring FRDM(2012) Nuclear Mass Data

This notebook explores the nuclear mass data from:

- **AME2020**: Atomic Mass Evaluation 2020 (experimental, 3,558 nuclides)
- **FRDM2012**: Finite Range Droplet Model 2012 (theoretical, 9,318 nuclides, Z=8–136)

Reference: Möller et al., Atomic Data and Nuclear Data Tables 109-110 (2016) 1-204

Key features:

- Complete superheavy element predictions up to Z=136
- 516 nuclides beyond currently synthesized elements (Z > 118)

In [1]:
```
import sys
sys.path.insert(0, '../src')

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

from hasna import NuclearDatabase

# Use DejaVu Sans for better Unicode support (β₂, subscripts)
plt.style.use('seaborn-v0_8-whitegrid')
plt.rcParams['font.family'] = 'DejaVu Sans'
%matplotlib inline
```

In [2]:
```
# Connect to database
db = NuclearDatabase()

# Show summary
summary = db.summary()
for key, value in summary.items():
    print(f"{key}: {value}")
```

```
ame2020_count: 3558
frdm2012_count: 9318
total_nuclides: 9420
both_exp_and_th: 3456
predicted_only: 5862
```

1. Nuclear Chart Overview

In [3]:
```
# Get all nuclides
nuclides = db.query("SELECT * FROM nuclides")
print(f"Total nuclides: {len(nuclides)}")
nuclides.head()
```

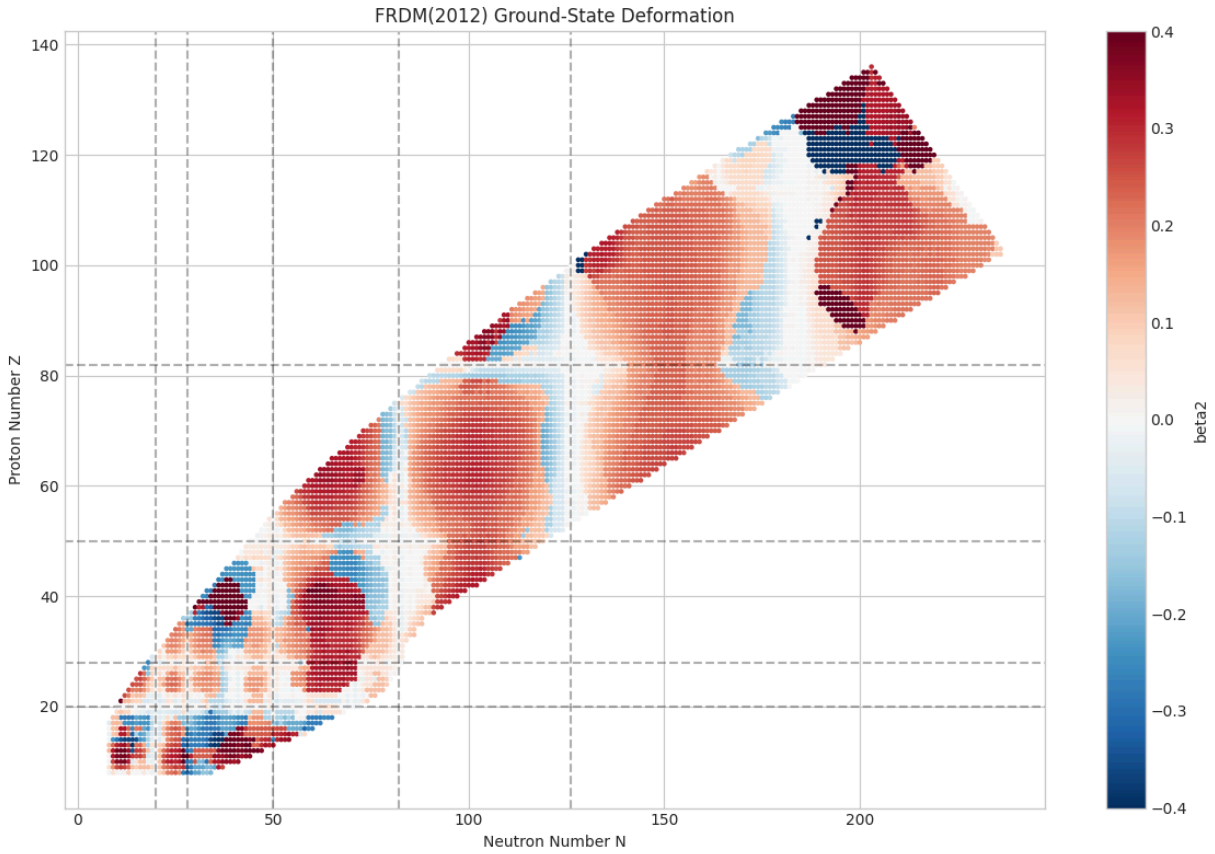
```
Total nuclides: 9420
```

Out[3]:

	Z	N	A	Element	mass_excess_exp_keV	mass_excess_exp_unc_keV	binding_pe
0	8	8	16	O	-4737.00217	0.00030	
1	8	9	17	O	-808.76421	0.00064	
2	8	10	18	O	-782.81634	0.00064	
3	8	11	19	O	3332.85800	2.63700	
4	8	12	20	O	3796.17200	0.88500	

In [4]:

```python
# Nuclear chart colored by deformation
fig, ax = plt.subplots(figsize=(12, 8))

frdm = nuclides[nuclides['has_theoretical']]
scatter = ax.scatter(frdm['N'], frdm['Z'], c=frdm['beta2'],
                     cmap='RdBu_r', s=5, vmin=-0.4, vmax=0.4)
plt.colorbar(scatter, label='beta2')

# Magic numbers
for z in [20, 28, 50, 82]:
    ax.axhline(y=z, color='k', linestyle='--', alpha=0.3)
for n in [20, 28, 50, 82, 126]:
    ax.axvline(x=n, color='k', linestyle='--', alpha=0.3)

ax.set_xlabel('Neutron Number N')
ax.set_ylabel('Proton Number Z')
ax.set_title('FRDM(2012) Ground-State Deformation')
plt.tight_layout()
```



2. Specific Nuclide Lookup

In [5]:
```
# Look up Fe-56 (most tightly bound nucleus)
fe56 = db.get_nuclide(z=26, n=30)
print("Fe-56 Properties:")
print(f"  Mass excess (exp): {fe56['mass_excess_exp_keV']:.1f} keV")
print(f"  Mass excess (th):  {fe56['mass_excess_th_keV']:.1f} keV")
print(f"  Difference: {fe56['exp_minus_th_keV']:.1f} keV")
print(f"  Deformation beta2: {fe56['beta2']:.3f}")
```

```
Fe-56 Properties:
  Mass excess (exp): -60607.2 keV
  Mass excess (th):  -60530.0 keV
  Difference: -77.2 keV
  Deformation beta2: 0.117
```

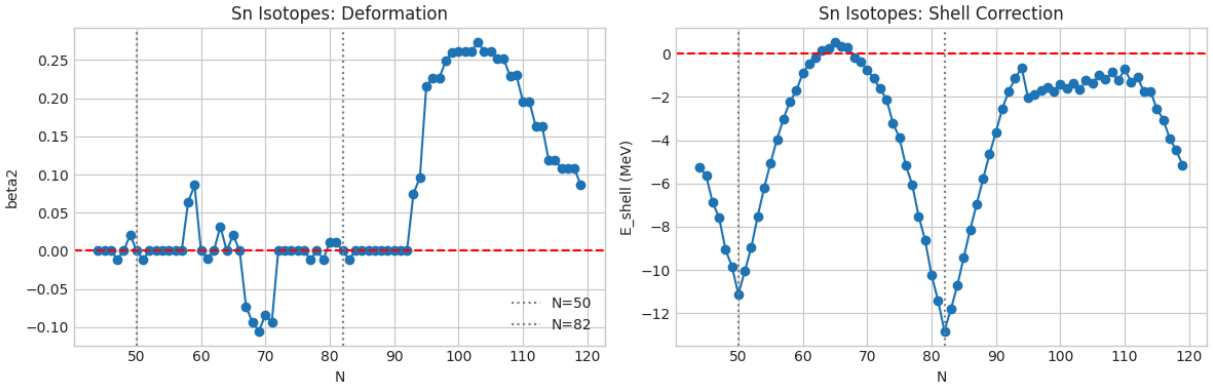
In [6]:
```
# Look up Pb-208 (doubly magic)
pb208 = db.get_nuclide(z=82, n=126)
print("Pb-208 Properties (doubly magic):")
print(f"  Mass excess (exp): {pb208['mass_excess_exp_keV']:.1f} keV")
print(f"  Mass excess (th):  {pb208['mass_excess_th_keV']:.1f} keV")
print(f"  Deformation beta2: {pb208['beta2']:.3f}  (spherical!)")
print(f"  Shell correction: {pb208['shell_pairing_MeV']:.2f} MeV")
```

```
Pb-208 Properties (doubly magic):
  Mass excess (exp): -21748.5 keV
  Mass excess (th):  -20920.0 keV
  Deformation beta2: 0.000  (spherical!)
  Shell correction: -13.93 MeV
```

3. Isotope Chains

In [7]:
```python
# Tin isotopes (Z=50, magic)
sn = db.get_isotopes(z=50)
print(f"Tin (Z=50) has {len(sn)} isotopes in the database")

fig, axes = plt.subplots(1, 2, figsize=(12, 4))

# Deformation
ax = axes[0]
ax.plot(sn['N'], sn['beta2'], 'o-')
ax.axhline(y=0, color='r', linestyle='--')
ax.axvline(x=50, color='gray', linestyle=':', label='N=50')
ax.axvline(x=82, color='gray', linestyle=':', label='N=82')
ax.set_xlabel('N')
ax.set_ylabel('beta2')
ax.set_title('Sn Isotopes: Deformation')
ax.legend()

# Shell correction
ax = axes[1]
ax.plot(sn['N'], sn['shell_pairing_MeV'], 'o-')
ax.axhline(y=0, color='r', linestyle='--')
ax.axvline(x=50, color='gray', linestyle=':')
ax.axvline(x=82, color='gray', linestyle=':')
ax.set_xlabel('N')
ax.set_ylabel('E_shell (MeV)')
ax.set_title('Sn Isotopes: Shell Correction')

plt.tight_layout()
```

```
Tin (Z=50) has 76 isotopes in the database
```

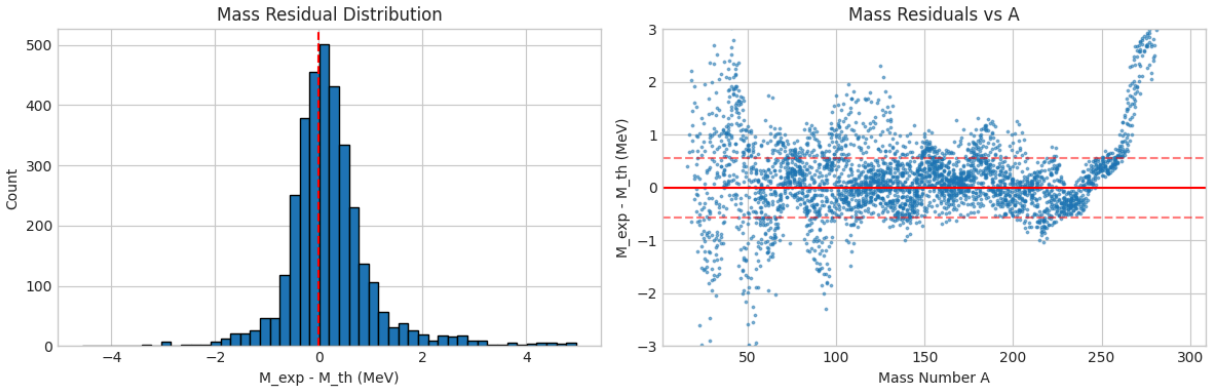


4. Experiment vs Theory

In [8]:
```
# Compare experimental and theoretical masses
compared = db.compare_masses(max_diff_keV=5000)
compared['diff_MeV'] = compared['exp_minus_th_keV'] / 1000

print(f"Nuclides with both exp and theory: {len(compared)}")
print(f"Mean difference: {compared['diff_MeV'].mean():.3f} MeV")
print(f"RMS difference: {np.sqrt((compared['diff_MeV']**2).mean()):.3f} MeV"
```

```
Nuclides with both exp and theory: 3436
Mean difference: 0.220 MeV
RMS difference: 0.894 MeV
```

In [9]:
```
fig, axes = plt.subplots(1, 2, figsize=(12, 4))

# Histogram
ax = axes[0]
ax.hist(compared['diff_MeV'], bins=50, edgecolor='black')
ax.axvline(x=0, color='r', linestyle='--')
ax.set_xlabel('M_exp - M_th (MeV)')
ax.set_ylabel('Count')
ax.set_title('Mass Residual Distribution')

# vs A
ax = axes[1]
ax.scatter(compared['A'], compared['diff_MeV'], s=3, alpha=0.5)
ax.axhline(y=0, color='r', linestyle='-')
ax.axhline(y=0.56, color='r', linestyle='--', alpha=0.5)
ax.axhline(y=-0.56, color='r', linestyle='--', alpha=0.5)
ax.set_xlabel('Mass Number A')
ax.set_ylabel('M_exp - M_th (MeV)')
ax.set_title('Mass Residuals vs A')
ax.set_ylim(-3, 3)

plt.tight_layout()
```



5. Deformed Nuclei

In [10]:
```
# Most deformed nuclei
deformed = db.get_deformed(min_beta2=0.35)
print(f"Highly deformed nuclei (|beta2| > 0.35): {len(deformed)}")
deformed[['Z', 'N', 'A', 'beta2', 'beta4']].head(10)
```

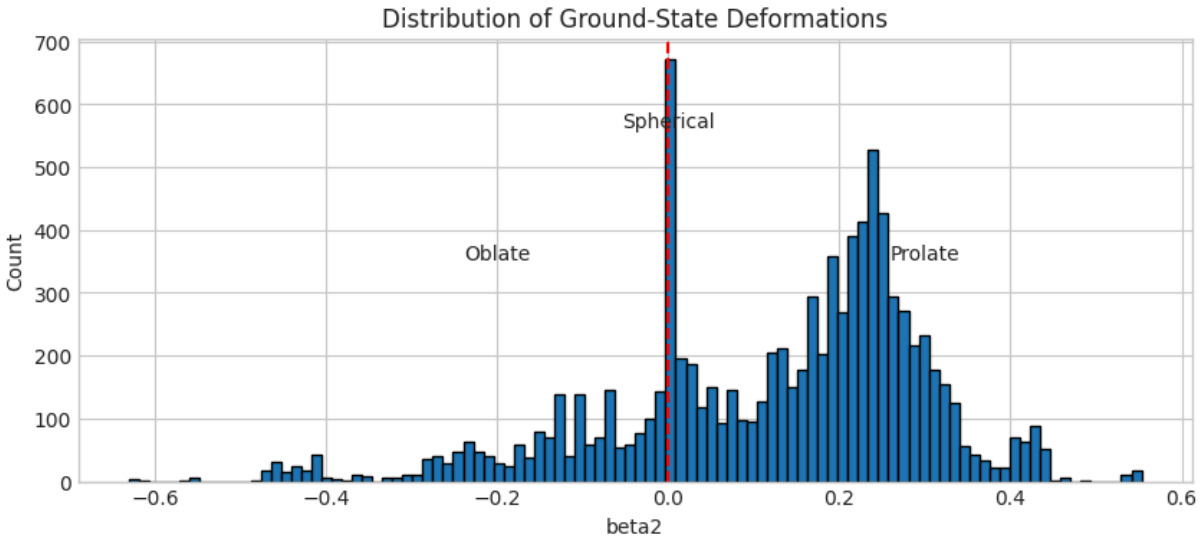
```
Highly deformed nuclei (|beta2| > 0.35): 624
```

Out[10]:

	Z	N	A	beta2	beta4
0	107	189	296	-0.630	0.154
1	108	189	297	-0.630	0.154
2	108	190	298	-0.629	0.156
3	105	187	292	-0.610	0.156
4	101	129	230	-0.566	0.107
5	102	130	232	-0.566	0.107
6	99	129	228	-0.557	0.104
7	101	128	229	-0.557	0.104
8	100	128	228	-0.557	0.104
9	100	129	229	-0.557	0.104

In [11]:
```
# Deformation distribution
frdm = nuclides[nuclides['has_theoretical']]

fig, ax = plt.subplots(figsize=(10, 4))
ax.hist(frdm['beta2'], bins=100, edgecolor='black')
ax.axvline(x=0, color='r', linestyle='--')
ax.set_xlabel('beta2')
ax.set_ylabel('Count')
ax.set_title('Distribution of Ground-State Deformations')

# Annotate
ax.annotate('Spherical', xy=(0, ax.get_ylim()[1]*0.8), ha='center')
ax.annotate('Prolate', xy=(0.3, ax.get_ylim()[1]*0.5), ha='center')
ax.annotate('Oblate', xy=(-0.2, ax.get_ylim()[1]*0.5), ha='center')
```

Out[11]:
```
Text(-0.2, 352.275, 'Oblate')
```



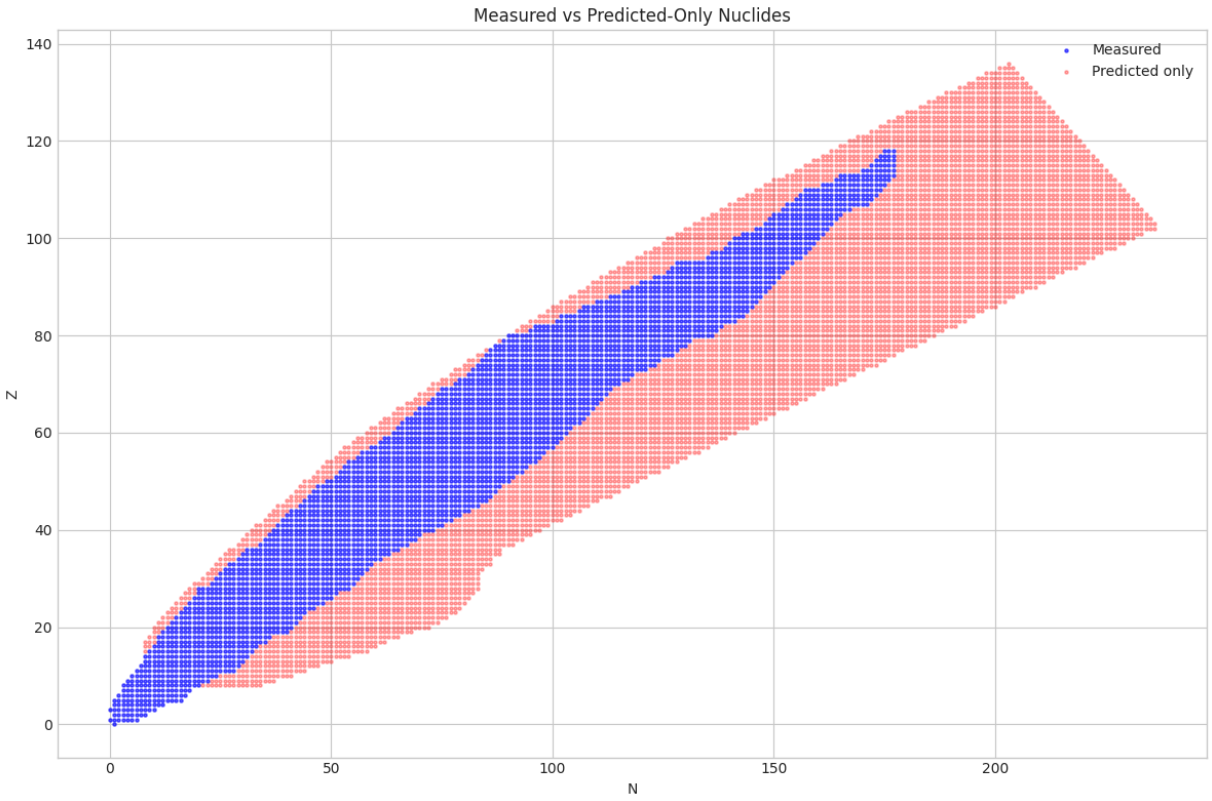
6. Predicted (Unmeasured) Nuclides

The FRDM2012 model provides predictions for many nuclides that have not been experimentally measured. This includes superheavy elements up to Z=136.

In [12]:
```
# Nuclides with only theoretical predictions
predicted = db.get_predicted_only()
print(f"Predicted-only nuclides: {len(predicted)}")
print(f"These are nuclides where FRDM2012 provides predictions,")
print(f"but no experimental mass measurement exists (as of AME2020).")
```

```
Predicted-only nuclides: 5862
These are nuclides where FRDM2012 provides predictions,
but no experimental mass measurement exists (as of AME2020).
```

In [13]:
```
# Where are the unmeasured nuclides?
fig, ax = plt.subplots(figsize=(12, 8))

# Measured
measured = nuclides[nuclides['has_experimental']]
ax.scatter(measured['N'], measured['Z'], c='blue', s=5, label='Measured', al

# Predicted only
ax.scatter(predicted['N'], predicted['Z'], c='red', s=5, label='Predicted or

ax.set_xlabel('N')
ax.set_ylabel('Z')
ax.set_title('Measured vs Predicted-Only Nuclides')
ax.legend()
plt.tight_layout()
```



7. Custom SQL Queries

In [14]:
```
# Example: Find all doubly-magic nuclei
magic_Z = [8, 20, 28, 50, 82]
magic_N = [8, 20, 28, 50, 82, 126]

doubly_magic = db.query(f"""
    SELECT Z, N, A, Element, beta2, shell_pairing_MeV,
           mass_excess_exp_keV, mass_excess_th_keV
    FROM nuclides
    WHERE Z IN {tuple(magic_Z)} AND N IN {tuple(magic_N)}
    ORDER BY A
""")
print("Doubly-Magic Nuclei:")
doubly_magic
```

```
Doubly-Magic Nuclei:
```

Out[14]:

	Z	N	A	Element	beta2	shell_pairing_MeV	mass_excess_exp_keV	mass_e
0	8	8	16	O	-0.010	-0.62	-4737.00217	
1	8	20	28	O	0.000	-2.87	52080.00000	
2	8	28	36	NaN	-0.322	-3.40	NaN	
3	20	20	40	Ca	0.000	-0.52	-34846.40200	
4	20	28	48	Ca	0.000	-1.94	-44224.86800	
5	28	20	48	Ni	0.000	-1.61	18178.00000	
6	28	28	56	Ni	0.000	-4.00	-53907.64900	
7	20	50	70	NaN	0.000	-4.43	NaN	
8	28	50	78	Ni	0.000	-5.89	-34880.00000	
9	50	50	100	Sn	0.000	-11.12	-57148.14600	
10	28	82	110	NaN	0.000	-8.20	NaN	
11	50	82	132	Sn	0.000	-12.82	-76546.55400	
12	82	126	208	Pb	0.000	-13.93	-21748.51900	

In [15]:
```
# Superheavy elements beyond current synthesis (Z > 118)
superheavy_unsynthesized = db.query("""
    SELECT Z, COUNT(*) as isotopes,
           MIN(N) as N_min, MAX(N) as N_max,
           AVG(beta2) as avg_beta2
    FROM nuclides
    WHERE Z > 118 AND has_theoretical
    GROUP BY Z
    ORDER BY Z
""")
print(f"Elements beyond Oganesson (Z=118): {len(superheavy_unsynthesized)}")
print(f"Total predicted nuclides (Z > 118): {superheavy_unsynthesized['isotc
print("\nIsotopes per element:")
superheavy_unsynthesized
```

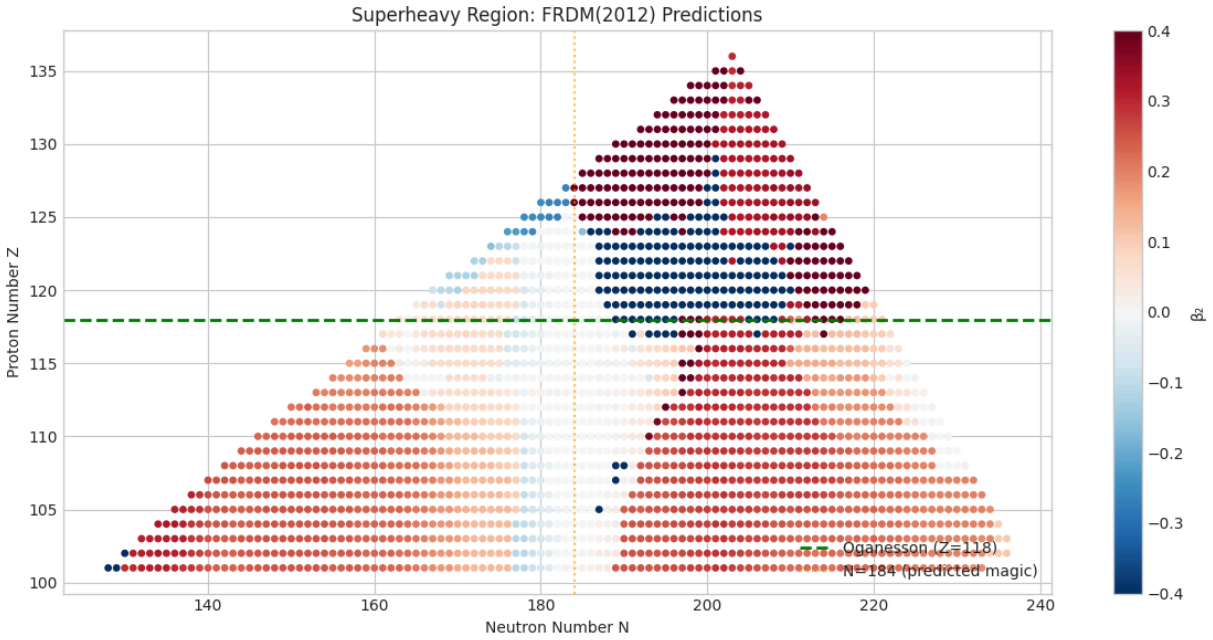
```
Elements beyond Oganesson (Z=118): 18
Total predicted nuclides (Z > 118): 516

Isotopes per element:
```

Out[15]:

	Z	isotopes	N_min	N_max	avg_beta2
0	119	56	165	220	-0.091054
1	120	53	167	219	-0.125925
2	121	50	169	218	-0.129620
3	122	46	172	217	-0.114696
4	123	43	174	216	-0.162535
5	124	40	176	215	0.015375
6	125	37	178	214	0.122622
7	126	34	180	213	0.279294
8	127	30	183	212	0.324667
9	128	27	185	211	0.374926
10	129	24	187	210	0.365125
11	130	21	189	209	0.385048
12	131	17	192	208	0.385706
13	132	14	194	207	0.382286
14	133	11	196	206	0.408818
15	134	8	198	205	0.385750
16	135	4	201	204	0.396500
17	136	1	203	203	0.296000

In [16]:

```
# Visualize superheavy region (Z > 100)
fig, ax = plt.subplots(figsize=(12, 6))

superheavy_all = db.query("SELECT * FROM nuclides WHERE Z > 100")
scatter = ax.scatter(superheavy_all['N'], superheavy_all['Z'],
                     c=superheavy_all['beta2'], cmap='RdBu_r',
                     s=15, vmin=-0.4, vmax=0.4)
plt.colorbar(scatter, label='β₂')

# Mark the boundary of synthesized elements
ax.axhline(y=118, color='green', linestyle='--', linewidth=2, label='Oganess

# Mark predicted "island of stability" region
ax.axvline(x=184, color='orange', linestyle=':', alpha=0.7, label='N=184 (pr

ax.set_xlabel('Neutron Number N')
ax.set_ylabel('Proton Number Z')
ax.set_title('Superheavy Region: FRDM(2012) Predictions')
ax.legend(loc='lower right')
plt.tight_layout()
```



In [17]:
```
# Close database when done
db.close()
```